Workgroup: COINRG
Internet-Draft:
draft-irtf-coinrg-use-case-analysis-02
Published: 4 December 2024
Intended Status: Informational
Expires: 7 June 2025

Authors: I. Kunze RWTH Aachen
J. Hong ETRI
K. Wehrle RWTH Aachen
D. Trossen Huawei
M. J. Montpetit Concordia
X. de Foy InterDigital Communications, LLC
D. Griffin UCL
M. Rio 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 challenges in defining general considerations. This document analyzes the use cases described in a COINRG companion document and potentially explores additional settings, to identify general aspects of interest across all use cases. The insights gained from this analysis will guide future COIN discussions.

Status of This Memo

Copyright Notice

Table of Contents

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 functionalities are 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 (Computing in the Network) fundamentally changes these observations by proposing to add meaningful compute functionalities within the network and thus between the end-hosts. However, building solutions for COIN-related problems is non-trivial, as there is currently no consensus on what exactly COIN is. In this context, [USECASES] provides a variety of use cases representing meaningful applications of COIN as well as a taxonomy to structure the descriptions of the different use cases. However, it does not provide further considerations; for example, it does not analyze the similarities of the different use cases and does not draw general conclusions.

This document was intended to fill that gap by performing an analysis of the use cases described in [USECASES] as well as additional ones. In light of the discontinuation of COINRG, progress on this document has stalled. Yet, to provide a stable reference for possible future analyses continuing the work started in this document, this document has been updated to the final set of use cases and research questions in [USECASES].

In the following, this document first introduces the terminology as defined in [USECASES] (Section 2) and the taxonomy used in [USECASES] for describing the use cases (Section 3). The rest of the document then provides the actual analysis, dividing the overall analysis into a few, more focused, smaller analyses.

2. Terminology

This document uses the terminology as defined in [USECASES] as listed below.

Programmable Network Devices (PNDs): network devices, such as network interface cards and switches, which are programmable, e.g., using P4 or other languages.

(COIN) Execution Environment: a class of target environments for function execution, for example, a JVM-based execution environment that can run functions represented in JVM byte code

COIN System: the PNDs (and end systems) and their execution environments, together with the communication resources interconnecting them, operated by a single provider or through interactions between multiple providers that jointly offer COIN capabilities

COIN Capability: a feature enabled through the joint processing of computation and communication resources in the network

(COIN) Program: a monolithic functionality that is provided according to the specification for said program and which may be requested by a user. A composite service can be built by orchestrating a combination of monolithic COIN programs.

(COIN) Program Instance: one running instance of a program

COIN Experience: a new user experience brought about through the utilization of COIN capabilities

3. COIN Use Cases Taxonomy

This document uses the use case taxonomy as defined in [USECASES] as described below. In particular, the objective of [USECASES] is to outline opportunities and propose possible research questions for consideration by the wider community when pushing forward the COIN vision. Furthermore, the individual use case descriptions aim to provide insights into the evolving solution space of collected COIN capabilities. As a result, [USECASES] defines the following taxonomy, which is used to describe each of the use cases:

1. Description: High-level presentation of the purpose of the use case and a short 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: Description of current methods that may realize the use case (if they exist), not claiming to exhaustively review the landscape of solutions.

4. Opportunities: An outline of how COIN capabilities may support or improve on the use case in terms of performance and other metrics.

5. Research questions: Essential questions that are suitable for guiding research to achieve the identified opportunities. The research questions also capture immediate capabilities for any COIN solution addressing the particular use case whose development may immediately follow when working toward answers to the research questions.

6. Additional desirable capabilities: Description of additional capabilities that might not require research but may be desirable for any COIN solution addressing the particular use case; we limit these capabilities to those directly affecting COIN, recognizing that any use case will realistically require many additional capabilities for its realization.

4. Analysis

The goal of this analysis is to identify aspects that are relevant across all use cases, thereby contributing to the 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 can provide valuable insights for both the future work of COINRG and the broader 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.

```
+--------------------------------------------------------------+
+                     Applicability Areas                      +
+ .............................................................+
+ Transport |   App  |    Data     |  Routing &  | (Industrial) +
+           | Design | Processing | Forwarding  |    Control   +
+--------------------------------------------------------------+

+--------------------------------------------------------------+
+    Distributed Computing FRAMEWORKS and LANGUAGES to COIN    +
+--------------------------------------------------------------+

+--------------------------------------------------------------+
+                ENABLING TECHNOLOGIES for COIN                +
+--------------------------------------------------------------+

+--------------------------------------------------------------+
+                      VISION(S) for COIN                      +
+--------------------------------------------------------------+
```

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.

In addition to these categories, there are use-case-specific research questions that are heavily influenced by the specific constraints and objectives of the respective use cases. These questions fall into the "applicability areas" category and can be further refined into the following subgroups:

- *Transport: Questions related to COIN's application, addressing the need to adapt transport protocols to handle dynamic deployment locations effectively.
- *App Design: Questions related to the design principles and considerations when developing COIN applications.
- *Data Processing: Questions related to the handling, storage, analysis, and processing of data in COIN environments.

*Routing & Forwarding: Questions that explore efficient routing and forwarding mechanisms in COIN, considering factors such as network topology, congestion control, and quality of service.

*(Industrial) Control: Questions specific to COIN's application in industrial control systems, addressing issues like real-time control, automation, and fault tolerance.

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). Note that we tackle several example domains in our applicability areas, e.g., regarding data processing Section 4.2.2.6, app design Section 4.2.2.5, or industrial control Section 4.2.2.8.

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, 4.2.7, 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 can end-hosts 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 4.2.7).

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, 4.1.4, 4.1.5, 4.1.8, 4.2.1, 4.2.4, 4.2.5, 4.2.6, 4.3.3, 5.2.1, 5.2.2, 5.2.3, 5.2.5, 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5, 6.1.5

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, RQ 4.2.1). In that case, how should this composition be done: manually or automatically? Additionally, one could opt for static composition once before deploying the program or one could enable dynamic recomposition at runtime (RQ 4.1.5). 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, RQ 6.1.5). 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 4.1.8, 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 4.1.4, 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) and how the frameworks will optimize the linking of different compute and communication resources (RQ 5.2.5). In this context, providing availability and accountability of resources can also be an important aspect as can be the

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 4.2.7). This is particularly challenging when such changes affect safety-related operations (RQ 4.3.3). 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 incorporate 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 due to the preliminary nature of this analysis.

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:

4.1.2, 4.1.3, 4.1.7, 4.2.6, 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 4.2.6). 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). Additionally, app designs also need to account for the limited capabilities of some of the possible COIN execution environments (RQ 4.1.2).
Many of these considerations boil down to a question of trade-off,

e.g, between storage and frequent updates (RQ 5.1.5), accuracy of control and implementation complexity (RQ 4.1.3), or how (new) COIN capabilities can be sensibly used for novel application design (RQ 5.1.1). Finally, finding automated solutions for (dynamically) deciding which trade-off option to choose could be a worthwile research endeavor as is finding out which information can be used for such decisions (RQ 4.1.7).

4.2.2.6. Applicability Area - Data Processing

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

3.2.4, 3.2.6, 4.2.2, 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), which parts of packet information can be used for these processing steps, e.g., payload vs. header information (RQ 4.2.2), or how different data streams can be combined (RQ 3.2.6, 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, 3.2.6, 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). These questions might become more challenging when multiple streams need to be considered and maybe even added or removed dynamically (RQ 3.2.6). 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:

3.1.9, 3.2.5, 3.3.1, 3.3.4, 4.1.1, 4.1.6, 4.1.8, 4.2.3, 4.3.1, 4.3.4

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 synchronization between these controllers (RQ 4.1.6) or, more generally, between different entities or flows/streams within the COIN system is also a relevant problem (RQ 3.1.9, RQ 3.3.4). A relevant question in this space is also how these different controllers interact (e.g., explicitly or implicitly) (RQ 4.1.8). 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), and how COIN software can prevent unsafe operations (RQ 4.3.4).

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

[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) pp. 1-6, 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-06, 12 September 2024, <https://datatracker.ietf.org/doc/html/draft-irtf-coinrg-use-cases-06>.

Authors' Addresses

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

Email: kunze@comsys.rwth-aachen.de

Jungha Hong
ETRI
218 Gajeong-ro, Yuseung-Gu
Daejeon
34129
Republic of Korea

Email: jhong@etri.re.kr

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

Email: wehrle@comsys.rwth-aachen.de

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

Email: Dirk.Trossen@Huawei.com

Marie-Jose Montpetit
Concordia University
Montreal

Canada

Email: marie@mjmontpetit.com

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

Email: xavier.defoy@interdigital.com

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

Email: d.griffin@ucl.ac.uk

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

Email: miguel.rio@ucl.ac.uk