A Distributed-Ledger-based Multi-Entity Cooperation Platform for Network-Cloud Recovery

Sugang Xu[1], Subhadeep Sahoo[2], Noboru Yoshikane[3], Sifat Ferdousi[2], Masaki Shiraiwa[1], Yusuke Hirota[1], Takehiro Tsuritani[3], Massimo Tornatore[4], Yoshinari Awaji[1], and Biswanath Mukherjee[2,5]
[1]National Institute of Information and Communications Technology (NICT), Tokyo, Japan, [2]University of California, Davis, USA, [3]KDDI Research, Inc., Saitama, Japan, [4]Politecnico di Milano, Milan, Italy, [5]Soochow University, Suzhou, China
E-mail: [1]{xsg, shiraiwa, hirota.yusuke, yossy}@nict.go.jp, [2]{subsahoo, sferdousi, bmukherjee}@ucdavis.edu, [3]{no-yoshikane, ta-tsuritani}@kddi.com, [4]{massimo.tornatore}@polimi.it

Abstract — **Cooperation among telecom carriers and datacenter (DC) providers (DCPs) is essential to ensure resiliency of network-cloud ecosystems. To enable efficient cooperative recovery in case of resource crunch, e.g., due to traffic congestion or network failures, we previously studied several frameworks for cooperative recovery among different stakeholders (e.g., telecom carriers and DCPs). Now, we introduce a novel Multi-entity Cooperation Platform (MCP) for implementing cooperative recovery planning, to achieve efficient use of carriers' valuable optical-network resources during recovery. We adopt a Distributed Ledger Technology (DLT) that ensures decentralized and tamper-proof information exchange among stakeholders to achieve open and fair cooperation. To support diverse types of cooperation, we develop a state machine representing the MCP operation and define state transitions associated to stakeholders' cooperation within the state machine. Moreover, we propose a signaling system in MCP to ensure simple and reliable state transitions for stakeholders during the cooperative recovery planning in large ecosystems. We experimentally demonstrate a proof-of-concept DLT-based MCP on a testbed. We showcase a DCP-carrier cooperative planning process, showing the flexibility of the proposed MCP to support diverse types of cooperation.**

Keywords—DLT, Open, Cooperation, Network-Cloud, Recovery.

I. Introduction

To accommodate the growing demands for 5G/6G services, telecom networks and datacenters (DCs) form large-scale network-cloud ecosystems (ecosystems, for short) hosting these services. Cooperation among telecom carriers (carriers, for short) and DC providers (DCPs) is essential to ensure the resiliency of these ecosystems and to prepare for and react to unexpected resource crunch caused by, e.g., traffic congestion, failures, man-made/natural disasters, etc. Cooperation is even more critical in large-scale ecosystems where the different infrastructures are owned by multiple entities. However, such cooperation is challenging, as different stakeholders may not be willing to disclose confidential information, such as network topologies and detailed resource availability.

We have proposed and modeled frameworks for DCP-carrier and carrier-carrier cooperation aided by a third-party entity, named provider-neutral exchange (PNE), and showed the benefits of confidentiality-preserving cooperative recovery [1],[2]. Figure 1 illustrates an example of DCP-carrier cooperation for disaster recovery aided by a PNE [1], with one DCP and two carriers. PNE can be a consortium of distributed co-location centers or Internet exchange points. Individual PNE nodes (packet routers/switches) interconnect different carriers, DCPs, and users in close proximity (e.g., in the same city). To conceal carriers' confidential information, e.g., optical network topology, damage information, etc., PNE can create a reference topology (public information) over PNE nodes. Then, carriers' optical networks can be abstracted to this public PNE reference topology for cooperation. Additionally, to optimize the cooperative recovery, PNE serves as a mediator among different carriers and DCPs for public information sharing (e.g., abstracted PNE reference topology and price), and for possible coordination between DCPs and carriers. The recovery plan is progressively improved with a sequence of computational subtasks undertaken by individual stakeholders. With cooperative recovery planning, DCPs' requests for carriers' connection services (e.g., a lightpath or IP-over-WDM connection) can be optimized by matching the resource availability in survived carrier optical networks. Simultaneously, the minimum set of necessary recovery tasks of carriers can be identified, achieving low-cost and fast recovery of ecosystems.

To realize such cooperative recovery planning and achieve the efficient use of carriers' valuable optical network resources, a cooperation platform among stakeholders is needed. Distributed Ledger Technology (DLT), such as blockchain [3], is a promising solution to ensure decentralized and tamper-proof information exchange among stakeholders (for more information, a survey on the application of DLT in 5G-and-beyond networks can be found in [4]). More recently, DLT-based management in multi-domain optical networks [5] and network-cloud systems [6],[7] has been investigated, showing the possibility of adopting DLT in cooperative resource allocation between operators for daily end-to-end network services. In this study, we design a novel DLT-based Multi-entity Cooperation Platform (MCP) to implement the open and fair cooperative recovery planning. This MCP is featured by (1) the definition of a flexible state machine that models diverse cooperation scenarios; (2) support for both public and non-public information sharing needed in cooperation; and (3) a broadcast-based signaling system for simplifying the large-scale cooperation. We demonstrate a proof-of-concept DLT-based MCP prototype. By defining the state transitions in the state machine according to the stakeholders' behaviors, we successfully showcase a *model-driven* DCP-carrier cooperative recovery planning. This shows the flexibility of MCP as a tool for supporting future diverse cooperation with low complexity.

This work is supported in part by US-Japan JUNO3 project: NSF Grant no. 2210384.

978-3-903176-62-1

The remainder of this paper is organized as follows. Sec. II presents our DLT-MCP design. Sec. III shows demonstration and experimental results. Sec. IV concludes the paper.

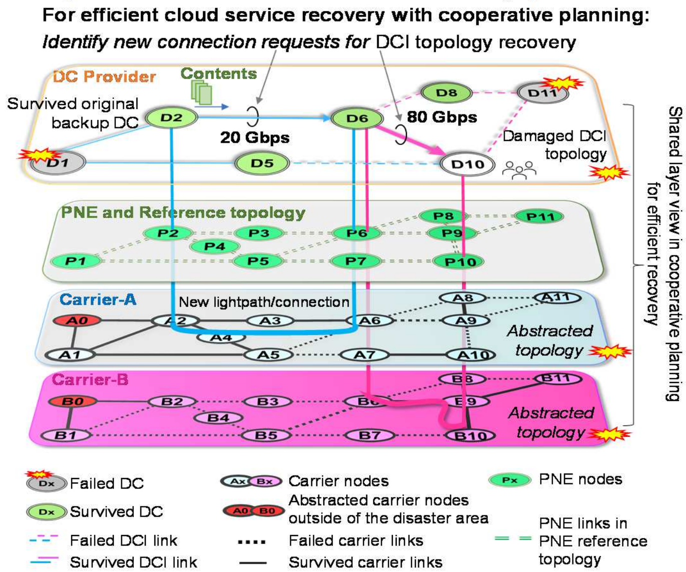

Fig. 1. Concept of cooperative network-cloud recovery with abstraction.

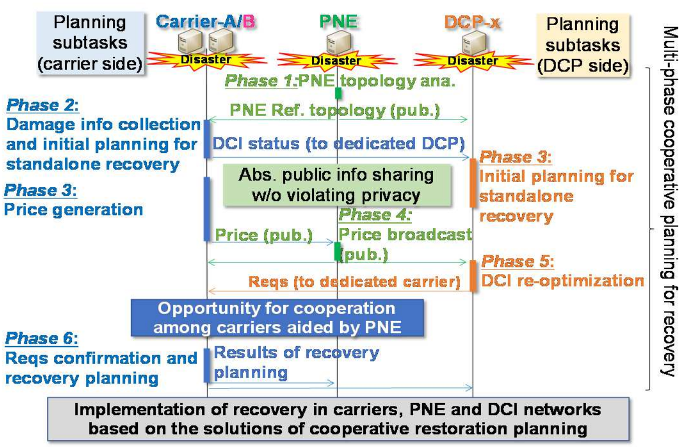

Fig. 2. Framework of multi-phase cooperative recovery planning.

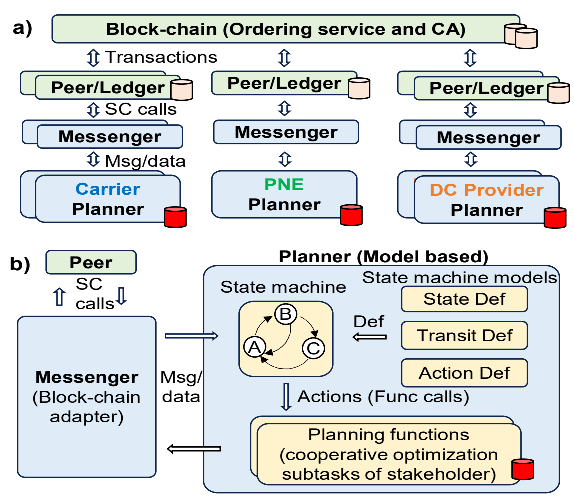

Fig. 3. Design of the DLT-based MCP to support cooperative recovery planning: (a) system structure; (b) building blocks of Planner supporting flexible model-driven cooperation.

II. DLT-based Multi-Entity Cooperation Platform

A. Framework of Cooperative Recovery Planning

Figure 2 shows an example use case to explain the design of our DLT-based MCP. It illustrates a multi-phase framework of DCP-carrier cooperative recovery planning [1], namely, among one DCP-x, two carriers (Carrier-A/B), and a mediator PNE. Note that, even though our example uses one DCP for sake of simplicity, multiple DCPs are allowed in cooperation. In cooperation, stakeholders are categorized by roles, i.e., carrier, DCP, and PNE. Each role has a collection of planning subtasks. These planning subtasks are arranged in six phases. In Phase 1, the PNE analyzes and broadcasts a PNE public reference topology. In Phase 2, Carriers A and B collect the damage information and perform the initial standalone recovery planning of their own optical networks. Then, Carriers A and B notify to their customers, e.g., DCP-x, the damage status of DC interconnection (DCI) links, which were established using the carrier's connection services before a disaster. In Phase 3, each carrier broadcasts the price (public information) of its connection services for each PNE node pair to DCP-x (also to the other carrier and the PNE, for possible carrier-carrier cooperation). A regular price is declared to offer a service over the survived resources, and an extra dummy (very high) price is declared to avoid the utilization of failed links over damaged resources. Simultaneously, DCP-x performs the initial standalone recovery planning by trying to reroute the traffic over the survived DCI links. In Phase 4, PNE aggregates and broadcasts the public price information of carriers. In Phase 5, having the price, DCP-x re-optimizes its DCI network topology by first using the survived connection services in carrier optical networks. Then, DCP-x delivers the new requests of connection services to Carriers A and B, respectively. In Phase 6, each carrier confirms the requests of DCP-x by performing the recovery planning, and notifies the results to DCP-x. Such cooperative planning poses special requirements on MCP which are described below.

(1) Flexible support for different stakeholders in diverse scenarios: During cooperation, various stakeholders behave differently depending on their roles, as shown in Fig. 2. Moreover, for different cooperation scenarios, e.g., DCP-carrier cooperation [1], carrier-carrier cooperation [2], and new types of cooperation in future study, etc., the sequence of information sharing and planning subtasks may be different. It is desirable that different stakeholders in diverse cooperation scenarios can be flexibly supported in an unified way by MCP.

(2) Different types of information sharing: The public information, e.g., PNE reference topology and price in Phases 1, 3, and 4 must be shared among all stakeholders in an open and fair fashion to ensure that all stakeholders have a common public information set in cooperation. On the contrary, non-public information, e.g., DCI status; DCP requests; and carrier confirmation in Phases 2, 5, and 6 should be exchanged in a closed manner between dedicated any carrier-DCP pair.

(3) Simple coordination mechanism for ensuring large-scale cooperation: In large-scale cooperation, the number of stakeholders will be large. A simple coordination mechanism among stakeholders is desirable to reduce the complexity in negotiations and avoid any scalability problem.

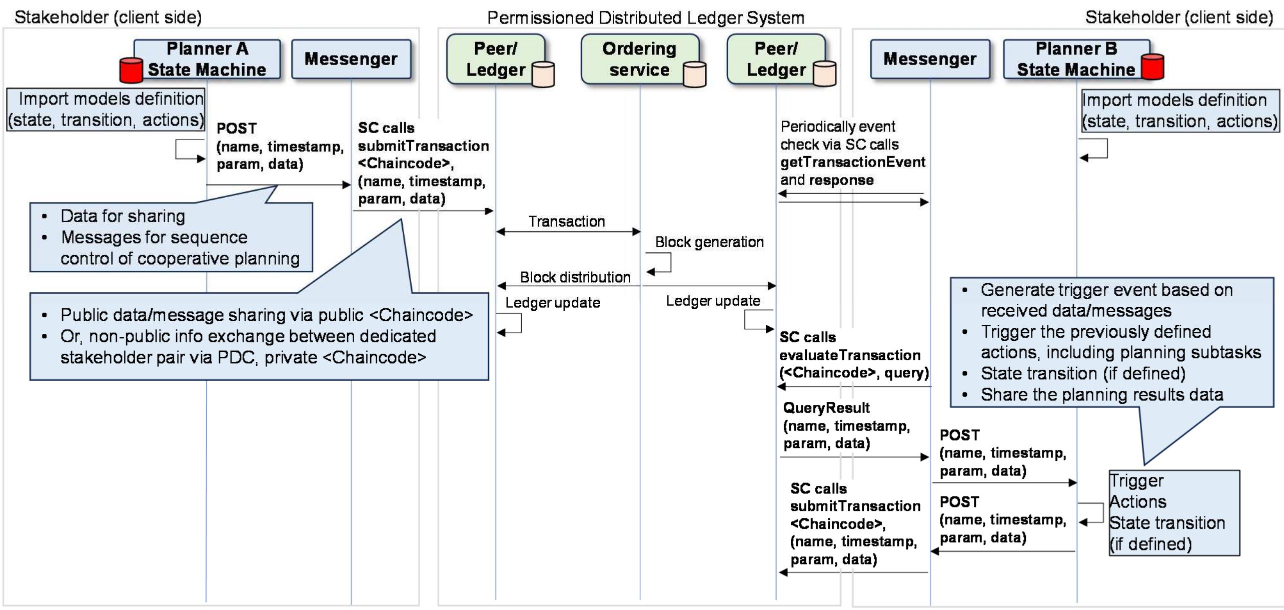

Fig. 4. Flow of information sharing and cooperation among stakeholders via a permissioned distributed ledger system.

B. *Flexible Definition of Cooperation with State Machine*

Figure 3(a) depicts the structure of our DLT-based MCP. Each stakeholder (such as carrier, DCP, and PNE) operates a Planner, i.e., a software module that implements a sequence of optimization subtasks in cooperation as mentioned in Sec. II.A. To support information sharing, a DLT subsystem including a Peer on each stakeholder site (green colored) is established among permissioned stakeholders. Each Planner is connected to its local Peer through a middleware element, called Messenger. Messenger offers to the Planner the capability of data/message sharing by calling the smart contracts (SCs) of the DLT subsystem (i.e., functions for information sharing) at the Peer and the data transactions in the DLT subsystem [3]. Figure 3(b) details the building blocks of the Planner. To flexibly define the different behaviors of stakeholders in diverse cooperation scenarios, we propose a common Planner design for all stakeholders. We adopt a State Machine as a core block of the Planner to control the behavior of individual stakeholders. We start by defining a block, called State Machine Model (SMM), to define the set of the state-machine components, namely, *states*, *transitions* (including condition rules), and *actions* of stakeholders. Since all stakeholders with the same role (e.g., all carriers) have the same behavior during cooperation, we only model and define the SMM for each role. Then, we can implement the desired behaviors of stakeholders based on the SMMs of their roles. For example, according to the multi-phase framework presented in Fig. 2, for each role, we define an initial *state* 0 before cooperation and six *states* corresponding to six phases during cooperation. In each *state*, upon receiving the trigger event (in the form of incoming data or messages) the Planner performs the defined *actions.* These *actions* include planning subtasks, such as carriers' planning/DCP requests confirmation, DCPs' planning/re-optimization, and data sharing, etc. When a trigger event is defined for *state transition*, the *state* is shifted. A novel *state-transition* mechanism supporting large-scale ecosystem is detailed in Sec. II.D. For supporting different cooperation scenarios, e.g., DCP-carrier and carrier-carrier cooperation, etc., we can redefine the SMMs and modify the *action* functions accordingly, fulfilling the cooperative planning in a *model-driven* fashion. SMM is demonstrated in Sec. III.

C. *DLT-Based Public/Non-Public Information Sharing*

Figure 4 details the information flow in MCP: (1) public and non-public information sharing and (2) the triggered *actions* among stakeholders via a DLT subsystem (e.g., only the permissioned members are involved). For example, in Phase 1, as shown in Fig. 2, after having imported the SMM in the state machine, the Planner of PNE (e.g., Planner A) starts to share the public reference topology through Messenger. The Messenger sends the formatted data (including name, timestamp, parameters, data, etc.) to the DLT subsystem by calling SC at the local Peer (e.g., with submitTransaction and a public <Chaincode> [8]). Such a public <Chaincode> can be treated as the group of all permissioned stakeholders and of all the functions for information sharing. Then, the public information is broadcasted, via transaction, block generation by the ordering service, and ledger update at individual Peers. Messengers periodically check the data by querying Peers (e.g., via getTransactionEvent in 2-second intervals). Upon receiving data (e.g., via evaluateTransaction and QueryResult), each Messenger notifies the data to the Planner (e.g., via a POST call). At the Planner (e.g., Planner B), a trigger event corresponding to the received data (e.g., an event "received the PNE reference topology") is generated and fed to the State Machine. Based on the current *state* of the state machine (e.g., Phase 1) and the trigger event, according to the *transition/action* definition in SMM, the State Machine calls the corresponding *actions* (e.g., processing PNE reference topology, etc.). If *state transition* is defined, the State Machine shifts to a new *state*, which is detailed in Sec. II.D. Consequently, Planner-B shares data (results) via the DLT subsystem. With respect to the non-public data exchange between dedicated stakeholder pairs, instead of the public <Chaincode>, private data collection (PDC) [8] (e.g., a private <Chaincode>) for the private information exchange among a limited number of stakeholders can be applied.

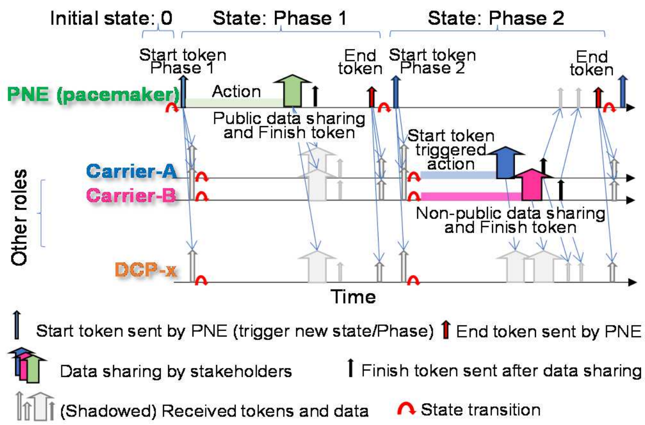

Fig. 5. PNE token-based state transision and actions among stekeholders.

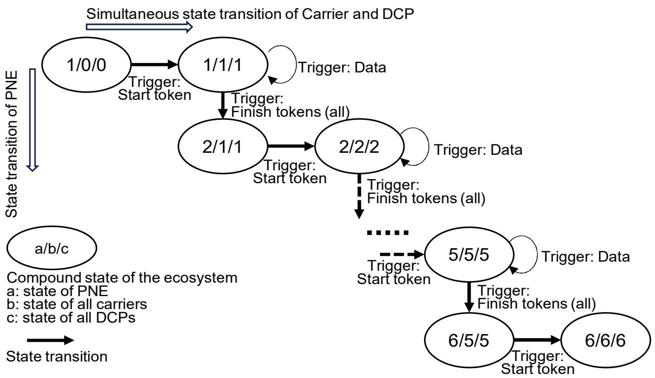

Fig. 6. Simplified global state transition aided by PNE and tokens.

D. Broadcast Token-Based Signaling System for Coordinated State Transition in Large-Scale Cooperation

We initially designed a set of conventional data-driven asynchronized *state transitions* for each stakeholder (e.g., independently triggered by the incoming data, such as PNE topology, price, requests, etc.). However, we found that, in the case of large-scale cooperation among many stakeholders, such an approach leads to complex *state transitions*, which are not only hard to debug, but also difficult for stakeholders to grasp the progress during the entire cooperative planning. To ensure and simplify the *state transitions* for all roles and to avoid the scalability problem, we propose a new broadcast-token-based signaling system to coordinate the *state transitions* of stakeholders. We introduce three broadcast tokens with the public-information-sharing capability. Namely, a Start and an End token are broadcasted by the PNE to explicitly signal the start and end of a phase for all the other stakeholders. And a Finish token is broadcasted by all stakeholders when they have completed the data sharing (in both public and non-public data sharing). Figure 5 illustrates this signaling system for the coordinated *state transition*, e.g., from Phase 1 to Phase 2 as shown in Fig. 2, which is described below:

State Phase 1: PNE starts by shifting (shown in Fig. 5 by a red turn arrow) from an initial *state* 0 to Phase 1, and performs a sequence of *actions.* First, PNE broadcasts a Start token (thin blue arrow) to explicitly signal the start of Phase 1. Second, PNE generates and shares the data (e.g., public PNE reference topology, represented as a wider arrow) followed by broadcasting a Finish token (thin black arrows) and an End token (thin red arrow), indicating the completion of its data sharing and the end of Phase 1, respectively. Upon receiving a Start token, Carrier-A/B and DCP-x are triggered to transition from *state* 0 to Phase 1 without *action*. When receiving the PNE reference topology data, Carrier-A/B and DCP-x are triggered to process the data without *state transition*.

State Phase 2: After sending the End token (at the end of Phase 1), PNE transits from Phase 1 to Phase 2 prior to the other stakeholders, and broadcasts a Start token to signal the start of Phase 2. Upon receiving this Start token, Carrier-A/B and DCP-x are triggered to first transit their *state* from Phase 1 to Phase 2. Second, Carrier-A/B execute two *actions* continuously: (*i*) each carrier performs its planning subtasks for damage evaluation and initial standalone recovery planning; and (*ii*) each carrier reports the evaluated DCI status (a wider arrow) to DCP-x via the non-public information sharing followed by broadcasting a Finish token. On the DCP-x side, since there is no planning subtasks of DCP in Phase 2, DCP-x bypasses the *action* on receiving Start token. When receiving DCI status reports from carriers, DCP-x is triggered to process the data without *state transition*. Back to the PNE side, the DCI status reports of carriers are not received due to the non-public information sharing, only the broadcast Finish tokens from carriers are received. After receiving all the Finish tokens from both carriers, PNE recognizes the condition of the end of Phase 2, i.e., all the carriers have completed their planning subtasks in Phase 2. Consequently, PNE is triggered to perform three *actions* sequentially: (*i*) broadcast an End token; (*ii*) transit its *state* from Phase 2 to Phase 3; and (*iii*) broadcast a Start token of Phase 3. This process is demonstrated in Sec. III.B. Such *state transition* and *actions* are continuously performed until they reach the last *state*, Phase 6, as shown in Fig. 2. If it is beneficial for stakeholders, the cooperative recovery plan will be implemented.

Figure 6 further depicts a global view of *state transitions* of all stakeholders in the ecosystem, showing the simplified *state transitions* in a large-scale cooperation. A circle denotes a compound *state* of the ecosystem during cooperative recovery planning. Labels *a*, *b*, and *c* represent the *states* of individual roles, PNE, carrier, and DCP, respectively. Upon receiving the Start token, all carriers and DCPs are simultaneously triggered to transition to new *states*, e.g., from 1/0/0 to 1/1/1, etc., and to perform their desired *actions* (e.g., planning subtasks if needed in a phase). By receiving Finish tokens from all the desired stakeholders, PNE transitions to new *states* prior to other stakeholders, e.g., from 1/1/1 to 2/1/1, etc. Upon receiving data, *actions* for data processing are triggered without *state transition*. With this "synchronized" *transition* (ignoring a short time lag in checking/receiving the tokens among stakeholders in the DLT subsystem), the space of the compound *state* in the ecosystem can be significantly reduced, resulting in low complexity of *transition*, which is easy to debug and understand. With PNE as a pacemaker, the design of state machines of all stakeholders is significantly simplified. Additionally, by taking advantage of the DLT guaranteed broadcasting, transmission of tokens (i.e., the signals of *transition*) is ensured, resulting in reliable *state transitions* in large-scale cooperation.

III. Demonstration

A. Open and Fair Information Sharing via DLT

Figure 7 shows the experimental setup of a MCP prototype (established over Linux servers), developed using a well-known

DLT platform, HyperLedger Fabric (HLF) [8], e.g., a set of Certificate Authorities (CAs), Orderers (nodes for offering the ordering service and block generation, etc.), and Peers. Pytransitions, a lightweight, object-oriented finite-state machine [9] was employed as the State Machine block in Planner. We demonstrate MCP with a *model-driven* DCP-carrier cooperative recovery planning (described in Sec. II.A) by demonstrating the SMMs of carrier, DCP, and PNE, including the token signaling system. The cooperative planning was successfully performed, showing the flexibility of MCP as a tool to facilitate the R&D of future diverse types of cooperation with low complexity.

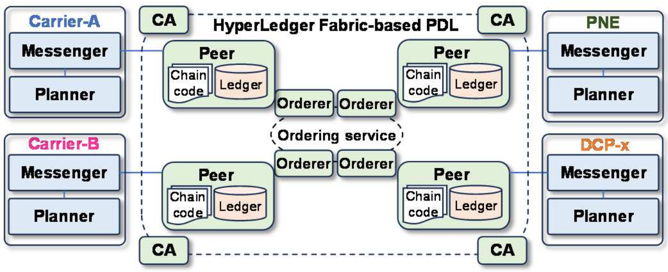

Fig. 7. Experimental setup with a HyperLedger Fabric-based testbed.

```
TransactioId: ff8d1431a38d2e3...
blockNo: 979
- arg:topology_broadcast_data
- arg:{"type":"topology","number":1,"from":"pne","to":"all"}
- arg:
    {
      "Phase": 1,
      "topology": {
        "uuid": "00000000-0000-0000-0000-000000000000",
        "node": [
          {
            "uuid": "00000001-0000-0000-0000-000000000000",
            "name": [
              {
                "value-name": "node-id",
                "value": "node1"
              }
            ],
            "node-edge-point": [
```

PNE broadcasted reference topology (public info sharing) via DLT system

PNE reference topology shared by PNE in Phase 1

```
TransactioId: ff8d1431a38d2e32353652e034142ccb92447c91
blockNo: 979
- arg:topology_broadcast_data
```

Trimmed Carrier-A log

```
TransactioId: ff8d1431a38d2e32353652e034142ccb92447c91
blockNo: 979
- arg:topology_broadcast_data
```

Trimmed Carrier-B log

```
TransactioId: ff8d1431a38d2e32353652e034142ccb92447c91
blockNo: 979
- arg:topology_broadcast_data
```

Trimmed DCP-x log

Fig. 8. Logs of DLT-based cooperative recovery planning in Phase 1: open and fair public information sharing ensured by the DLT subsystem.

We first tested an instance of open and fair public information sharing among stakeholders. Figure 8 shows the public PNE reference topology sharing from PNE to all stakeholders in Phase 1, which was logged at the Messengers of all stakeholders. We can see that, with a public <Chaincode>, all stakeholders received the same public information, i.e., with the same transaction ID and block number. The PNE public reference topology is shown in the PNE logs, and is omitted in the logs of Carrier-A/B and DCP-x due to space limitation. The public price information sharing performed in Phases 3 and 4 was observed in the same way (but not shown due to space limitation). It is guaranteed by the HLF that all the shared public information in all the Peers and permissioned stakeholders is consistent and tamper-proof. This demonstrates the DLT-based open/fair information sharing in cooperation, a key requirement in cooperative recovery planning. Namely, no stakeholder is able to monopolize and manipulate the important information in recovery (e.g., locations for applying the survived connection service of carriers) without the majority knowing it. The non-public information exchange, SMM, broadcast token-based signaling system aiding *transition* are shown in Sec. III.B.

B. *Model-Driven Cooperation and Broadcast Token-Based Signaling System for Simple State Transition*

```
transitions = [
 ≈
  {'trigger':'rcv_start_token', 'source':'phase1', 'dest':'phase2',
     'after' : 'planning_standalone_and_snd'
  },

  {'trigger':'rcv_dci_status_data', 'source':'phase2', 'dest':'phase2',
     'prepare' : 'rcv_dci_status'
  },

  {'trigger':'rcv_finished_token', 'source':'phase2', 'dest':'phase3',
     'prepare'    : 'rcv_finished',
     'conditions' : 'chk_all_finished',
     'before'     : 'snd_end',
     'after'      : 'snd_start',
  },
 ≈
```

Start token triggered actions and state transition of **carrier** and **DCP**

Data (DCI status report) triggered action of **DCP**

Finish token triggered actions and transition of **PNE**

Fig. 9. Example of SMM (definition of state transitions and behaviors of carrier, DCP, and PNE roles in Phase 2).

In SMM, we defined an initial *state* 0 and six *states* corresponding to six phases in cooperation, and a collection of *state transitions* for modelling the state machines (behaviors) of stakeholders depending on their roles. Figure 9 shows a part of SMM for defining three *state transitions* rules of carrier, DCP, and PNE roles from Phase 1 to Phase 2 as an example. Each rule consisted of a trigger, the current *state* (source), the next *state* (dest), and a sequence of *actions*. In Fig. 10, we present the corresponding logs of all stakeholders collected at the Planners, demonstrating the *state transitions* and *actions* of stakeholders.

1) Transition/actions on receiving Start token (by Carrier/DCP)

The first block in Fig. 9 defines a trigger event "rcv_start_token" (i.e., generated by the Planner on receiving a Start token) and the *state transition* from the *state* Phase 1 to Phase 2. Upon receiving a Start token sent by PNE (see Fig. 10, row Seq. 1), since Carrier-A/B and DCP-x were in state Phase 1, this rule was applied, and they simultaneously transited to Phase 2 accordingly. In this rule, we defined an "after" *action*, namely, "planning_standalone_and_snd". After *state transition*, Carrier-A/B performed their initial standalone recovery planning, and sent the DCI status reports dedicated to DCP-x via PDC [8] (as explained in Sec. II.C) followed by broadcasting a Finish token indicating the completion of data sharing (see Fig. 10, rows Seq.2 to 4, Carrier-A/B logs). As there was no planning subtask of DCP in Phase 2, DCP-x executed this *action* without processing. As PDC was used for DCI status reports, "dci_status_data" was only received by DCP-x (see Fig. 10, rows Seq.2 and 3, DCP-x log). Note that, before sending Start token, the PNE had shifted to Phase 2 (see Fig. 10, row Seq.1, PNE log), this rule was not applied by PNE.

2) Action on receiving Data via PDC (by DCP)

The second block in Fig. 9 defines a trigger event "rcv_dci_status_data" (i.e., on receiving a DCI damage report) and a "prepare" *action*, namely, "rcv_dci_status" without *state*

Seq	PNE-aided State Transition	Carriers' triggered Data Sharing and Actions		DCPs' triggered Actions
	PNE log	Carrier-A log	Carrier-B log	DCP-x log
1 Start token	[pne] **do Func:"snd_start" on State:phase2** [pne] send message "start_token" "name": "start_token", "param": { "from": "pne", "to": "all"	[crr_a] recieved message "start_token" "name": "start_token", "param": { "from": "pne", "to": "all"	[crr_b] recieved message "start_token" "name": "start_token", "param": { "from": "pne", "to": "all"	[dcp_x] recieved message "start_token" "name": "start_token", "param": { "from": "pne", "to": "all"
2 Actions		[crr_a] **do Func:"planning_standalone_and_snd" on State:phase2**	[crr_b] **do Func:"planning_standalone_and_snd" on State:phase2**	[dcp_x] **do Func:"planning_standalone_and_snd" on State:phase2 (dcp_x passes)**
3 PDC data sharing, and Actions		[crr_a] send message "dci_status_data" "name": "dci_status_data", "param": { "from": "crr_a", "to": "dcp_x"}, "data": [{"Phase": 2, "dci-status": **Carrier-A and -B performed non-public dedicated information exchange to DCP-x, reporting DCI damage information (dci_status_data) via PDC.**	[crr_b] send message "dci_status_data" "name": "dci_status_data", "param": { "from": "crr_b", "to": "dcp_x"}, "data": [{"Phase": 2, "dci-status": [	[dcp_x] recieved message "dci_status_data" "name": "dci_status_data", "param": { "from": "crr_b", "to": "dcp_x"}, "data": [{"Phase": 2, "dci-status": [[dcp_x] **do Func:"rcv_dci_status" on State:phase2** [dcp_x] recieved message "dci_status_data" "name": "dci_status_data", "param": { "from": "crr_a", "to": "dcp_x"}, "data": [{"Phase": 2, "dci-status": [[dcp_x] **do Func:"rcv_dci_status" on State:phase2**
4 Finish tokens	[pne] recieved message "finished_token" "name": "finished_token", "param": { "msg_name": "dci_status_data", "from": "crr_a", "to": "all" [pne] recieved message "finished_token" "name": "finished_token", "param": { "msg_name": "dci_status_data", "from": "crr_b", "to": "all"	[crr_a] send message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_a", "to": "all" [crr_a] recieved message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_b", "to": "all"	[crr_b] send message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_b", "to": "all" [crr_b] recieved message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_a", "to": "all"	[dcp_x] recieved message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_a", "to": "all" [dcp_x] recieved message "finished_token" "name": "finished_token", "param": { "msg_name":"dci_status_data", "from": "crr_b", "to": "all"
5 Action	pne] **do Func:"rcv_finished" on State:phase2[** pne] **do Func:"chk_all_finished" on State:phase2** [pne] condition : OK			
6 End token	[pne] **do Func:"snd_end" on State:phase2** [pne] send message "end_token" "name": "end_token", "param": { "from": "pne", "to": "all"	[crr_a] recieved message "end_token" "name": "end_token", "param": { "from": "pne", "to": "all"	[crr_b] recieved message "end_token" "name": "end_token", "param": { "from": "pne", "to": "all"	[dcp_x] recieved message "end_token" "name": "end_token", "param": { "from": "pne", "to": "all"
7 Start token	[pne] **do Func:"snd_start" on State:phase3** [pne] send message "start_token" "name": "start_token", "param": { "from": "pne", "to": "all"	[crr_a] recieved message "start_token"	[crr_b] recieved message "start_token"	[dcp_x] recieved message "start_token"

Fig. 10. Logs of PNE-aided state transitions and actions triggered by tokens (i.e., Start, End, Finish tokens) and data sharing in Phase 2.

transition. Upon receiving "dci_status_data" from individual carriers, DCP-x triggered *action* "rcv_dci_status" to collect the DCI status information from Carrier-A and -B, respectively (see the highlighted part in Fig. 10, row Seq.3, DCP-x log).

3) Transition/actions on receiving Finish tokens (by PNE)

The third block in Fig. 9 defines a trigger event "rcv_finished_token" (i.e., generated on receiving the Finish tokens) and *state transition* from *state* Phase 2 to Phase 3, for PNE. Four *actions*, namely, "prepare", "conditions", "before", and "after", were defined for (1) processing the received Finish tokens; (2) checking the "condition" of *state transition,* if all carriers have completed the data sharing with Finish tokens; (3) broadcasting an End token to signal the end of Phase 2; and (4) broadcasting a new Start token to signal the start of the next Phase 3, respectively. These *actions* and *state transition* of PNE at the end of Phase 2 were logged (see Fig. 10, rows Seq.5 to 7, PNE log). Other roles bypassed these process.

Such token-based *state transitions* and triggered *actions* are continuously performed for other phases, e.g., price sharing and so on, until the last *state*, Phase 6, is reached. The running time of the six-phase *transition* (excluding the time for planning subtasks) was less than 3 min which was acceptable. The corresponding logs are omitted due to space limitation.

Note that, in Phase 3, the public price information sharing was performed after the price analysis by carriers, in which the starting time of price sharing might be different and unfair for carriers. By extending the signaling system, it is possible to further synchronize the sharing of public information among stakeholders to enhance the fairness in cooperation. For other cooperation scenarios, e.g., carrier-carrier cooperation and other types of cooperation in future, we can redefine the SMMs and modify the *action* functions, fulfilling cooperation in a *model-driven* fashion. These are envisioned as future work.

IV. Conclusion

We proposed and demonstrated a DLT-based model-driven multi-entity cooperation platform to support the cooperative planning of network-cloud ecosystem recovery. Cooperative planning in future diverse cooperation can be flexibly performed in an open and fair fashion. With a signaling system introduced for coordinating stakeholders, simple and smooth cooperation among large number of stakeholders can be achieved.

References

[1] S. Sahoo, et al., "Datacenter-carrier cooperation over optical networks during disaster recovery," in Proc. OFC2022, W2A.23, 2022.

[2] S. Xu, et al., "A novel strategy of carrier cooperation with coordinated scheduling for swift failure/disaster recovery," in Proc. ONDM2023, pp. 163–168, 2023.

[3] S. S. Shetty, et al., Blockchain for Distributed Systems Security. Wiley-IEEE Press, 2019.

[4] M. Tahir, et al., "A review on application of blockchain in 5G and beyond networks: taxonomy, field-trials, challenges and opportunities," IEEE Access, vol. 8, 2020.

[5] S. Fichera, et al., "Leveraging blockchain to ratify QoT performance in multi-domain optical networks," in Proc. ECOC2019.

[6] P. Alemany, et al., "Evaluation of the abstraction of optical topology models in blockchain-based data center interconnection," J. Opt. Commun. Netw., 14, 4, pp. 211–221, 2022.

[7] R. Vilalta, et al, "End-to-end inter-domain transport network slice management using DLT-enabled cloud-based SDN controllers," in Proc. OFC2023, Tu3D.3, 2023.

[8] HyperLedger Fabric, [Online] https://www.hyperledger.org/

[9] Pytransitions, [Online] https://github.com/pytransitions/transitions