ネットワーク災害訓練のシナリオ記述コストを低減するインターフェイスの設計と実装

柏崎 礼生[1,a)] 西内 一馬[2,b)] 北口 善明[3,c)] 市川 昊平[4,d)] 近堂 徹[5,e)] 中川 郁夫[1,6,f)]
菊池 豊[7,g)]

概要：情報システムの動作を業務プロセスに含む事業において，その事業継続計画をより現実に即した内容として改善させるためには定期的な訓練が必要である．訓練は多様な障害を模倣する実際上のシナリオが必要である．大規模化，広域化し，複数の組織からなる分散システムにおいてはこのシナリオの作成コストが無視できない．本稿では複数の組織がネットワークで接続された広域分散システムにおけるネットワーク防災訓練のシナリオの作成コストを低減させるインターフェイスの設計と実装を説明し，その効果を評価する．

Design and implementation of interface to reduce costs of describing scenario for network disaster training

Hiroki Kashiwazaki[1,a)] Kazuma Nishiuchi[2,b)] Yoshiaki Kitaguchi[3,c)] Kouhei Ichikawa[4,d)]
Kondo Tohru[5,e)] Ikuo Nakagawa[1,6,f)] Yutaka Kikuchi[7,g)]

Abstract: To improve a business continuity plan to more realistic one, it is necessary for the business that include processes of information communication technology system to conduct trainings periodically. Trainings needs their virtual scenarios that emulate evaluate various faults. The costs to generate scenarios can not be ignored in large scale, wide area distributed system that consists of many, various organizations. This paper shows designs and implementations of interfaces to reduce generating costs for network disaster trainings. Evaluations of effectiveness of the interface are also discussed.

Keywords: disaster training resilience cost evaluation

[1] 大阪大学
Osaka University
[2] 株式会社シティネット
City Net Inc.
[3] 金沢大学
Kanazawa University
[4] 奈良先端科学技術大学院大学
Nara Institute of Information Science and Technology
[5] 広島大学
Hiroshima University
[6] 株式会社インテック
Intec Inc.
[7] 高知工科大学
Kochi Insitute of Technology
[a)] reo@cmc.osaka-u.ac.jp
[b)] nishiuchi@city-net.jp
[c)] kitaguchi@imc.kanazawa-u.ac.jp
[d)] ichikawa@is.naist.jp
[e)] tkondo@hiroshima-u.ac.jp
[f)] ikuo@inetcore.com
[g)] yu@kikuken.org

1. 背景と目的

分散システムとはネットワーク上に配置された計算機が互いにメッセージのやりとりによって通信し，連携するソフトウェアシステムである [1]．具体的な例として電話網や携帯電話網，インターネット，ワイヤレスセンサーネットワーク，WWW，P2P ネットワークなどが挙げられ，現在の生活に必要不可欠な基盤を支えるシステムとなっている．例えば広域分散システムを代表するインターネットについて，2015 年の日本におけるインターネットのブロードバンドサービス契約者の総ダウンロードトラフィック

は 5.4Tbps，総アップロードトラフィックは 1.1Tbps と推定されている[*1]．全世界規模のインターネットトラフィックは 2016 年で平均 34Tbps に及び[*2]，今後 5 年間で 3 倍以上となることが予測されている．インターネットトラフィックの増大をもたらした要因の一つに携帯電話の普及が挙げられる．2014 年における世界の携帯電話普及率は 96.3%であり，日本や北米・欧州以外の地域における 2000 年と 2014 年の携帯電話の普及率を比較すると 21.7 倍の差 (契約数 2.6 億から 57.1 億) が生じている[*3]．特に途上国においては固定通信網の整備より移動体通信網の整備が進んでいるケースがある．移動体通信網の整備がより多くの顧客を獲得することができるためである．

2016 年 4 月 14 日以降，熊本県と大分県で発生し続けている熊本地震では 4 月 14 日および 16 日に震度 7 を記録し，停電および伝送路断を原因として通信キャリアによる通信サービスが利用できなくなる世帯が発生した．これに対して NTT DOCOMO は熊本県阿蘇郡南阿蘇村，熊本県阿蘇市の無線基地局 (計 4 局) を除き[*4]4 月 20 日 20 時 59 分に地震前のサービス状態に復旧させた．地震発生後，停電により 78 局，伝送路断により 6 局がサービス断状態となったが，衛星移動基地局車，移動電源車や発動発電機の運用により迅速な復旧を実現している[*5]．ソフトバンクは係留気球無線中継システム 1 局，移動基地局車 4 台，中継伝送路を確保するための衛星通信設備 16 局，可搬型基地局 16 局，可搬型基地局 12 台，発電機 12 台を設置して地震による障害に対応し，NTT DOCOMO と同じく 27 日に全域において地震発生前と同等に復旧した[*6]．KDDI (au) のみが復旧手法について情報を公開していないが，国内 3 キャリア内では最も早い 26 日に地震発生前と同等に復旧している[*7]．ITmedia Mobile の記事によると車載基地局 8 台，可搬基地局 5 局，移動電源車 12 台，ポータブル発電機 45 台が配備されたという[*8]．

地震に限らず台風がもたらす大雨や高潮，それによりもたらされる土砂災害など 2016 年に発生した自然災害は多様である．これらの自然災害に対して予測をすることはできたとしても予防[*9]をすることはできない．著者らは減災 (disaster mitigation) の取り組みを推進している．減災の意義は図 1 により説明される．減災の手段を講じなかった場合，自然災害の被害に遭った人々の人生の品質 (あるいはインフラストラクチャに限って言えばインフラストラクチャがユーザに提供する品質) は自然災害の発災とともに低下する．低下した品質は被害者本人あるいはその周囲の取り組みにより発災前の品質へと復旧する．減災は発災前の訓練や情報提供，インフラストラクチャの増強などにより発災直後の品質減を緩和する．また発災後の復旧も事前の訓練により効率化され，発災以前の品質へと復旧する時間が短縮される．すなわち発災により発生する三角形の間隙の面積が縮小する．減災により縮小された三角形と，減災が講じられなかった場合の三角形の間にある領域が，減災がもたらす意義として説明することができる．

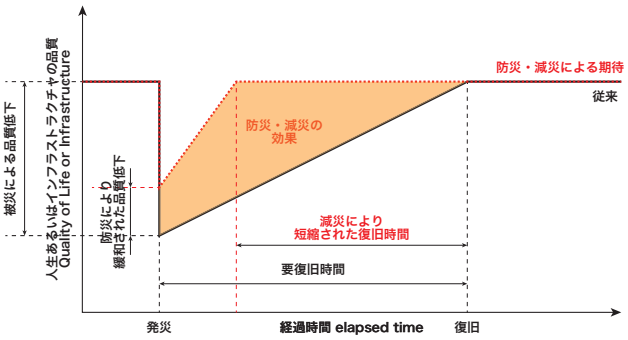

図 1 減災の効果
Fig. 1 Effectiveness of disaster mitigations

2012 年 8 月に内閣府が発表した「南海トラフの巨大地震による津波高・浸水域等 (第二次報告) 及び被害想定 (第一次報告)」によると，想定されるケース「『四国沖』に『大すべり域+超大すべり域』を設定」および「『四国沖〜九州沖』に『大すべり域+超大すべり域』を設定」において高知県幡多郡黒潮町および土佐清水市は最大津波高 (満潮位・地殻変動考慮) において国内最大の 34m と推定されている[*10]．高知県では県内の高等学術機関 5 組織が協働し，

*1 総務省による報道資料「我が国のインターネットにおけるトラヒックの集計・試算」2016 年 3 月 2 日
http://www.soumu.go.jp/menu_news/s-news/01kiban04_02000103.html
*2 Cisco Systems Inc.:“Cisco Visual Networkin Index”, June 1, 2016
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html
月あたり予測値 88.7EB から算出した．
*3 総務省: “平成 28 年度情報通信白書”
http://www.soumu.go.jp/johotsusintokei/whitepaper/h28.html
*4 これらの地域が立ち入り禁止区域であったため．最終的にこれら 4 局も 4 月 27 日には復旧した．
*5 報道発表資料 平成 28 年熊本地震からの復旧状況について＜2016 年 4 月 28 日＞
https://www.nttdocomo.co.jp/info/news_release/2016/04/28_00.html
*6 ソフトバンク株式会社 プレスリリース 平成 28 年熊本地震の影響に伴うソフトバンクならびにワイモバイル携帯電話サービスの復旧について
http://www.softbank.jp/corp/group/sbm/news/press/2016/20160425_02/
*7 KDDI ホーム 熊本県熊本地方を中心とした地震の影響について
http://news.kddi.com/important/news/important_20160426443.html
*8 熊本地震，KDDI (au) のカバーエリアが復旧
http://www.itmedia.co.jp/mobile/articles/1604/26/news138.html
*9 悪い事態の起こらないように前もってふせぐこと．(デジタル大辞泉より)
*10 南海トラフの巨大地震による津波高・浸水域等 (第二次報告) 及び被害想定 (第一次報告) 資料 1-2 都府県別市町村別最大津波高一覧表＜満潮位＞
http://www.bousai.go.jp/jishin/nankai/taisaku/pdf/1_

高知 IX，高知 PoP，南国 PoP を連携したさせた冗長構成でインターネットや相互接続を実現している．既存の ICT システムに意図的に障害を起こすことにより，システムの冗長性や障害対策の機能および ICT 関係者間での連絡体制等を確認・検討し，実際に機能する事業継続計画を策定することを目的として，TEReCo4[*11] プロジェクトは 2013 年度にネットワーク防災訓練を行った[*12]．この取り組みでは様々な障害要因をロジックモデル手法で可視化しており，3 つの障害パターンでの検証が行われた．その結果，本来不通になるはずの障害パターンにおいてもインターネットへの導通が確認されたことで運用者が把握していない冗長構成の存在が発覚するなど，防災訓練の実施により，耐災害性・耐障害性を向上させるのみならず，本来の目的以外の効果が現れた事例が報告された [2,3].

高知県でのネットワーク防災訓練ではネットワーク障害の発生は人為的に，かつ実際に人間の手による手動操作で行われたものである．しかし例えば組織内においてネットワークトラフィックの少ない時間帯である深夜から朝にかけての時間帯において障害を発生させようとすると，必然的に人力で訓練のための障害を発生させるコストを要する．前述の取り組みでも障害がもたらす影響を加味して，1 月 5 日の午前 5 時から訓練を開始しているため，定期的に行うことは困難であることが考えられる．我々は，この訓練が定常的に高い頻度で，しかも多様な障害で行われることが必要であると考えている．人力で行うためには多様な障害シナリオを記述するコストが必要であり，なおかつ高い頻度で障害を発生させ，その評価を行い，検証するコストを算出すると，人力で実現することは現実的ではない．また先の検証においては複数箇所で同時に発生する障害を人為的に作り出すことや，障害後のネットワーク情報を収集すること，そして障害発生後に元の状態へ戻すことの困難さを明らかにしている．

情報システムの中でも特に分散システムは防災訓練を行うために複数のステイクホルダーの了承を得ることが求められるために困難であったが，それと同時に高い耐災害性・耐障害性が求められるシステムでもある．このような動機付けのもと，障害を形式的に記述し，かつ障害時にネットワーク情報を収集することが可能であり，評価後にネットワーク状態を元に戻すことが可能な，分散システムの耐災害性・耐障害性の検証・評価・反映を行うためのプラットフォーム “DESTCloud” が開発された．

2. DESTCloud の設計と実装

本章において，障害発生プラットフォームを実現するために必要なネットワーク障害の分類と実現に向けた課題を整理し，実装手法を明記する．

2.1 障害パターンの分類

障害パターンには自然災害に起因する障害から装置故障等に起因する障害まで，多様な要因が考えられる．また，影響範囲がどの程度か，あるいは空間的な変化を伴うか否か，時間的な推移をどのように表現するかについても検討が必要となる．本節では，総務省「大規模災害等の緊急事態における通信 確保の在り方に関する検討会」[*13]で示されている災害事象や「情報通信ネットワーク安全・信頼性基準」[*14]の内容をもとに，災害時における通信設備等に対する障害に焦点を絞り，各事象に対してネットワーク装置に適用する制御について検討した．通信障害は主に 2 つの原因に大別されると考えられる．ひとつはトラフィック集中による輻輳，もうひとつは回線や機器等のハードウェア・設備障害である．これらをより細かな区分に分類し，それぞれの障害要因と具体的な症状の対応付けを行い，各々について本プラットフォームで実装する機能についてまとめた (図 2).

発生区分	障害要因	症状	実装する機能
制御・運用・ソフトウェア	通信規制制御	輻輳	遅延発生＋n% パケロス
			トラフィックシェープ
	不正な経路伝搬	経路ループ	RIB/FIB 強制書換
		経路フラップ	
		経路障害（宛先不達）	
ネットワーク機器	装置故障（全体）	通信断（全体）	インターフェイスダウン
	装置故障（部分）	通信断（部分）	
	リソース過負荷	パケットロス	n% パケロス
		遅延増大	遅延追加
通信回線	拠点間通信ケーブル断	通信断（部分）	インターフェイスダウン＋100% パケットロス
	中継器・交換機故障		
	トラフィックの集中	輻輳	遅延発生＋n% パケロス
			トラフィックシェープ
設備環境	局舎損壊	通信断（全体）	インターフェイスダウン＋100% パケットロス
	電源喪失		
	空調故障		
		通信断（部分）	

図 2 DESTCloud におけるネットワーク障害の分類
Fig. 2 Classification of network disorder in DESTCloud

このように，例えば通信断であっても，ネットワーク機器自体が故障する場合と中継機・交換機が故障する場合では，実際の通信でみられる症状が異なることが予想される．

2.2 要求要件

前節で述べたような障害パターンを実ネットワーク上で実現するために要求される要件をまとめる．

2.pdf

[*11] Traffic Engineering for Regional Communities, version 4

[*12] 福本昌弘ら「災害時に事業継続性を発揮する情報通信インフラのための運用計画改善手法および冗長化技術の研究開発」総務省地域 ICT 振興型研究開発 (平成 25 年度〜26 年度, 四国総合通信局)
http://www.soumu.go.jp/main_content/000284013.pdf

[*13] 総務省「大規模災害等緊急事態における通信確保の在り方に関する検討会」
http://www.soumu.go.jp/main_sosiki/kenkyu/saigai

[*14] 総務省「情報通信ネットワーク安全・信頼性基準」
http://www.soumu.go.jp/menu_seisaku/ictseisaku/net_anzen/anshin

課題 1: プログラム可能で拡張性のある障害発生の実現
障害に対するシステムの挙動を客観的かつ正確に分析するためには，想定する障害発生状況に応じて時間的・空間的に変化する故障・トラブルを正確に何度でも再現可能であることが必要である．また，生成する障害発生の状況・条件を柔軟に変更可能とし，多角的な検証を可能とする拡張性も備えることも求められる．したがって，障害発生機構はプログラムで記述可能としたプログラム可能性を具備する必要がある．プログラムで指定した通りに何度でも同じ状況で障害を生成可能とし，また，条件を変化させて様々なバリエーションに応じた障害を記述可能とする．

課題 2: 実ネットワークにおける適用可能性 耐障害性の検証のためには，机上における訓練のみでは不十分であり，実環境での定期的かつ継続的な検証が耐災害性を確保するための重要な要素である．そのため，提案するシステムは実際のネットワークにおいて適用可能である必要がある．ゆえに実際の個々のネットワーク装置をプログラムでコントロールする仕組みが必要である．加えて，障害発生時の状況を迅速に分析可能とするために，個々のネットワーク装置からの情報を自動で集約し，分析者に提供する必要がある．

課題 3: 実ネットワークの定常運用への迅速な復帰 耐障害性の検証は定常的かつ継続的に検証すべきであるが，その広域分散システムが提供するサービスの可用性・利便性の毀損は最小限に留める必要がある．ゆえに検証終了後は迅速に定常運用状態へ復帰する必要がある．このような機能は，障害発生の前後におけるシステムの挙動の比較・検証にも有用性が高い．

2.3 設計

DESTCloud では先に挙げた課題のうち，プログラム可能な障害発生を実現するために SDN (Software Defined Networking) 技術を活用する．SDN 技術は，物理的な制約に縛られることなく柔軟なネットワーク構築を実現する仕組みである．本提案ではプログラム可能なネットワーク障害を繰り返し発生させることにより，再現性を確認できる．これにより障害による問題の発生が偶発的なものであるのか，必然的に発生し得るものかを区別することができる．

Cisco Systems 社が提供する SDN プラットフォームである onePK (One Platform Kit) を用いて実装した．onePK は同社のオペレーティングシステムである Cisco IOS 等に対し，C，Java，Python 等の言語で操作できる API を提供し，ネットワーク管理者は CLI を想定したパーサーを用意することなく，API を使って柔軟にネットワーク機器を制御することができる．onePK ではすべての通信制御を SDN で行うのではなく，従来型のネットワーク設定によるネットワーク構成が可能である点が特徴であり，一時的に優先度を高めたプロセスをソフトウェア制御で投入できる．ゆえに実行プロセスを終了することで定常運用の状態に容易に復旧でき，複雑な多発的障害の注入時においてもその復旧に対する完全性を担保することが可能である．onePK には，Cisco IOS に搭載されているイベントマネージャ機能である EEM (Embedded Event Manager) の API も提供されており，障害発生プラットフォームにおけるネットワーク機器の統合的な状態収集を行うことができる．EEM では SNMP や Syslog に代表されるイベント検出技術を扱う事ができ，ネットワーク内に発生している状態を時系列で把握できるため実効性に富む．

図 3 に，障害発生プラットフォームのアーキテクチャを示す．本プラットフォームでは，災害等で発生する様々な障害をレイヤ構造で実現する．システム管理者や災害訓練実施者による訓練シナリオの入力が行われる．この訓練シナリオの入力ユーザインターフェイス (User Interface: UI) は，Web ブラウザを通して対話的に訓練シナリオを構築する方法のほか，後述する災害シナリオコントローラ (Disaster Scenario Controller: DSC) が解釈可能な YAML 形式のシナリオを書いてコマンドラインインターフェイス (Command Line Interface: CLI) でプログラムを用いて登録する方法もある．ゆえに人間の手でなくとも訓練シナリオの記述と登録を行うことができる．この訓練シナリオを解釈し，指定された時間から訓練を開始するスケジューラの役割を担うのが DSC である．DSC は訓練シナリオに基づく障害を実際のネットワーク機器へ注入し，解除を行うために network entities の制御のために統一的な API を用意する．これは，従来より一般的に利用されている SNMP [4] や NETCONF [5] などの非 SDN 技術を組み合わせることを想定しているためで，この API により制御手法毎の差異を吸収することができる．

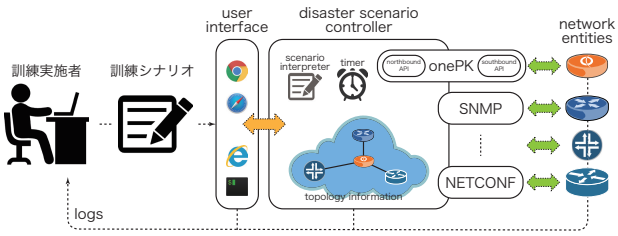

図 3 DESTCloud の模式図
Fig. 3 Diagram of DESTCloud

2.4 実装と構築

2015 年に行われた初歩的な実装では，図 2 で定義したネットワーク障害のうち，“インタフェースダウン”，“100%パケットロス” および “経路表強制書き換え” という 3 つの障害を実装した．“インタフェースダウン” と “100%パケットロス” は，それぞれルータの “shutdown コマンド” と

“ACL による 100%パケットロス” と同様の制御を行う実装としている．また，“経路表強制書き換え” は，CLI における “static route の追加” と同様の制御を行うことで，動的経路に対して administrative distance が小さい経路を設定し，経路の上書きを実現している．以下に，実装したコマンドとその引数を記す．

インタフェースダウン

```
linkctl.py {down|up} <router> <ifid>
```

100%パケットロス

```
pktloss.py {100|0} <router> <ifid>
```

経路表強制書き換え

```
dcroute.py {add|dell}\
<static_route> <router> <ifid>
```

これらのコマンドは制御用サーバにて実装しており，各障害を CLI で実行可能としている．そのため，複数の障害を時系列に発生させる障害シナリオをプログラムとして記述することができる．さらに，再現性を有する障害処理を繰り返し実行することが可能となる．図 4 に，構築した障害発生プラットフォーム構成を示す．このプラットフォームは，JGN-X[*15] 上に配置された onePK 対応ルータを利用して構築し，札幌，仙台，大手町，名古屋，大阪，岡山，広島，福岡の各アクセスポイント (AP) で利用可能とした．配置されているルータの構成は拠点毎で異なっており，大手町 AP および大阪 AP の 2 カ所が Cisco ASR9006，その他の AP では Cisco ASR1004 となっている．このような JGN-X の環境を利用し，全国 5 拠点のユーザセグメント (広島大，高知工科大，大阪大，NAIST，金沢大) とルータを結ぶ論理パスをそれぞれ構築し，複数の論理パスで各ルータを結ぶネットワークとして構築した．金沢大のみ JGN-X への直接の接続性を持たないため，SINET4[*16] による L2VPN サービスを経由した接続形態としている．

今回のプラットフォームではユーザセグメント以外に管理セグメントも用意している．これは，onePK の制御用セグメントでもあり，オペレータが操作する制御用サーバから API 経由で各ルータと制御メッセージがやり取りされる．本セグメントはフラットな L2 で構成されており，サーバから各ルータに対して API 接続する際は TLS 認証が必須となる．なお，制御用サーバは高知工科大に設置しており，ここから検証プラットフォーム全体の制御を行う構成としており，各ネットワーク機器からのログ情報を収集するサーバも併設している．

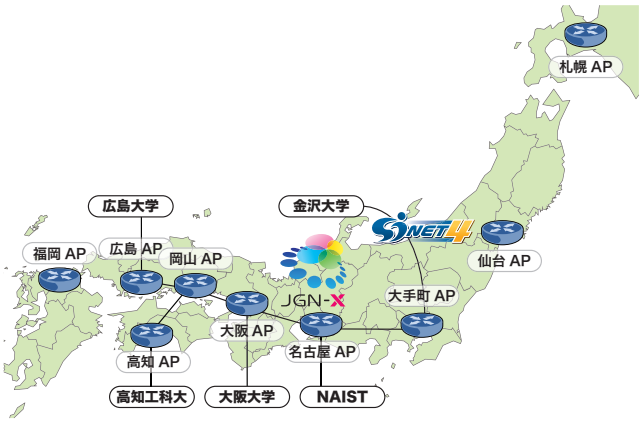

図 4 JGN-X と SINET4 を用いた DESTCloud の論理パス構成
Fig. 4 Logical topology of DESTCloud on JGN-X and SINET4

2.5 評価実験

我々が提案する障害発生プラットフォームの有効性を明らかにするために，評価実験を行った．評価手法として，実際の広域分散システムにおける耐障害性検証を実施し，提案手法の有効性を評価する．

JGN-X を用いて構築した障害発生プラットフォーム上に，評価対象の広域分散システムを配置して検証環境を準備した．評価用の広域分散システムとしては，株式会社インテックが開発した分散ストレージシステムである EXAGE/Storage を広域分散対応したシステムを用いている．検証環境のネットワークは金沢大と広島大，奈良先端大の 3 拠点にて EXAGE/Storage による広域分散ストレージを構成しており，それぞれの拠点は BGP の経路交換で接続するネットワーク環境としている．BGP の keepalive 時間は一般的な値[*17]の 1/3 に，EXAGE/Storage はデフォルトの設定値を利用している．各拠点では，EXAGE/Storage を NFS マウントするサーバを CentOS 6.6 にてそれぞれ用意し，TCP による NFSv3 経由による書き込み／読み出し性能を計測し評価できる構成としている．

スプリットブレイン問題は，EXAGE/Storage のように複数ノードを相互接続したクラスタシステムで発生しうる問題で，切り離されたグループ間の同期ができないことによるデータベースの競合や一意性の喪失が発生する問題である．EXAGE/Storage では，スプリットブレイン問題への対処として，多数派のグループ (サイト) を継続利用し，少数派を強制停止 (フェンシング) させるといった一般的な手法を実装している．多数派・少数派の判断は，コアサーバ内に設定するキャッシュサーバの合計台数にて判断している．キャッシュサーバは複数拠点に配置しており，システム全体で奇数台稼働している．各拠点のサーバは，

[*15] http://www.jgn.nict.go.jp
[*16] http://www.sinet.ad.jp
[*17] http://www.janog.gr.jp/doc/janog-comment/bcop-ebgp.txt

表 1 各拠点における書き込み・読み出し性能
Table 1 Write performance and read performance at each site

拠点	書き込み性能	読み出し性能		
		金沢データ	広島データ	奈良データ
金沢	59.5 MB/s	63.8 MB/s	65.9 MB/s	62.6 MB/s
広島	56.5 MB/s	65.9 MB/s	94.4 MB/s	87.0 MB/s
奈良	50.5 MB/s	54.9 MB/s	59.5 MB/s	66.8 MB/s

これらのキャッシュサーバとの通信確認を keepalive にて実施しており，過半数との通信が維持できなくなった場合に自身のクラスタが少数派であると判断し動作を停止する．通信不能と判断する条件は，キャッシュサーバに対する keepalive が 3 回続けて失敗した場合としており，「スプリットブレイン検出 keepalive 時間」の 3 倍の時間が経過した段階でフェンシング処理が開始される．

今回の評価実験では，この広域分散対応を実施した EXAGE/Storage を用い，拠点間障害による影響を確認する実験を行う．検証プラットフォームによる評価に先立ち，各拠点からの書き込み性能と読み出し性能を dd コマンドで計測した．書き込みと読み出しをそれぞれ 12 回づつ実施し，最大値と最小値の除いたデータの平均値として算出した．以後の評価実験における書き込みおよび読み出し性能の計測に関しても，定常時の値と比較するために，障害を発生させた後に実施した 12 回の計測結果を元に算出している．表 1 に計測結果を示す．読み出しに関しては，自拠点および他拠点のファイル読み出しの比較のため，すべての拠点に対して相互に実施している．書き込み性能は，コアサーバの台数と相関がある．読み出し性能は金沢拠点を除いて自拠点で生成したファイルの読み出しスループットが高い値を示す．これは，ファイル作成時に生成されるメタデータが書き込みを実施した拠点のコアサーバ上に作られる実装による影響である．金沢拠点に関してはこの影響が観測される前に性能の上限に達している．

拠点間の通信断によるスプリットブレイン問題を発生させ，システムとしての挙動を確認する評価実験を行った．広島 AP がダウンすることで広島拠点が切り離される障害を想定し，広島 AP にある ASR に対してインタフェースダウンの障害を発生させる．今回の構成では，スプリットブレインの検出に用いるキャッシュサーバを 7 台設定し，金沢拠点に 3 台，広島と奈良拠点に 2 台づつ配置している．上記の障害により，金沢拠点と奈良拠点が接続される側が多数派となることから，広島拠点のコアサーバ群がフェンシング対象となる．

検証の結果，少数派となる広島拠点がフェンシング処理により強制停止されることが確認でき，広島拠点が切り離されている状態においても，広島拠点で作成されたファイルを読み出すことができた．これは，ブロックデータの多重度が 3 であることから，すべての拠点においてレプリケーションデータを保持していることに起因している．また，障害復旧後にフェンシングからの復旧を実施した際のデータベース競合も発生することなく，読み出しが可能である．金沢拠点から広島拠点で作成したデータを読み出す性能は，49.5 MB/s (定常時:65.9MB/s) に低下した．読み込み中に障害を発生させた場合，読み出し処理に 120 秒を要するケースもあったが読み出し不能とはならなかった．拠点間通信断による耐障害性検証を同じ障害シナリオにて 3 度繰り返し計測したうち 1 回において対象ファイルが読み出し不能となる事象が確認された．この現象は現在の実装において想定外の挙動であったことから，詳細なログ解析とコード解析が必要となった [6].

2.6 設計の問題点

2015 年に行われた評価実験を経て DESTCloud の設計の問題を把握することができた．前述の評価実験では訓練実施者と DSC との間の UI は CLI であった．そのため訓練実施者は訓練を行うネットワークのトポロジ，ネットワーク機器 (ルータ) の IP アドレス，ルータ間を接続するインターフェイスの ID をその都度参照して訓練シナリオを作らなければならなかった．そのため 1 つの訓練シナリオを作るのに要する時間が膨大になり，繰り返しの訓練を行うことはできるものの，作ったシナリオを参照し派生シナリオを作るといった再利用は困難であった．

3. DESTCloud UI の改善

訓練実施者がルータの IP アドレス，制御に必要な情報 (プロトコル，ポート番号) を知っていることは求められなければならない．しかしルータ同士が，どのインターフェイスを使ってどのように接続されているかについては訓練実施者の認識と実態が乖離している可能性もある．間違った情報をもとに訓練シナリオを記述した場合，本来操作すべきではないインターフェイスに影響を与える可能性がある．実態に即した情報を訓練実施者に与え，その中の情報を取捨選択させることにより訓練シナリオを記述させることが UI の役割であるとした．

3.1 設計と実装

UI，DSC，およびネットワーク機器を制御するプログラム (Network faults implementation: NFI) の設計および具体的な実装とその連携を図 5 に示す．

訓練実施者は訓練を実施する広域分散システムを構成するネットワーク機器の情報を UI に登録する．UI はこの登録された情報から YAML 形式[*18]のネットワーク機器情報 (entities YAML) を DSC に登録する．DSC はこのデータを受け取ると，UI に受領確認の ACK を返す．DSC はこ

[*18] http://yaml.org/spec/history/2001-05-26.html

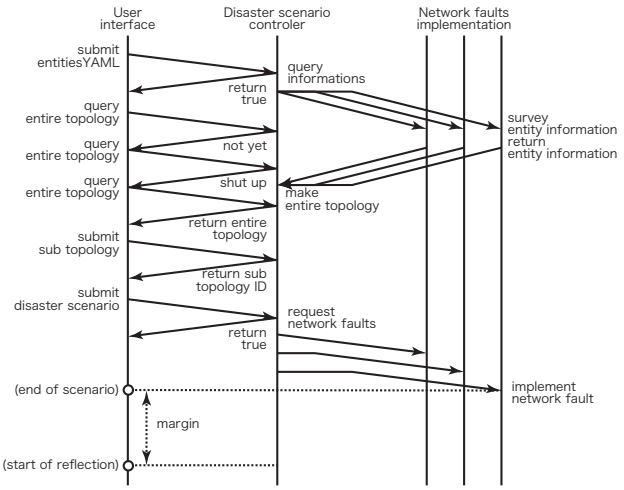

図 5 DESTCloud の通信様式図
Fig. 5 A communicatin diagram of DESTCloud

のネットワーク機器の情報をもとに，NFI に対してネットワーク情報の問い合わせを行う．NFI はネットワーク機器ごとに存在し，ネットワーク機器の各ポートに設定された L3 ネットワーク情報を DSC に対して返答する．これらの情報をもとに DSC は UI 部から提供されたネットワーク機器がどのようなトポロジで接続されているかを知ることができる．UI は DSC に対してこのトポロジの問い合わせを繰り返す．DSC が NFI から得た情報によりネットワーク機器全体のトポロジ (entire topology) を生成し終えると，この問い合わせに対してトポロジを記述した YAML を返答する．訓練実施者は entire topology を見て，訓練で使用する資源を選択する．この資源はネットワーク機器全体のトポロジに対する部分トポロジなので，sub topology と称する．訓練実施者は UI を通して sub topology を指定すると，UI は DSC に sub topology を登録する．DSC はその sub topology を内部データベースに登録し，この sub topology の ID を UI に返答する．訓練実施者はこの sub topology 内で実装する障害とその発生時間を，災害訓練開始時間からの相対時刻で指定する．全ての障害を記述し終えたら訓練シナリオと開始時間を DSC に登録する．DSC はこの訓練シナリオを解釈し，指定された相対時間に指定されたネットワーク機器に対して指定された障害を発生するよう NFI と通信を行う．災害シナリオに記載された全ての障害を実装し終えた時が災害シナリオの終了であり，訓練実施者はこのあと一定の時間を margin として指定し，その時間が経過するまで収集されたログを，この災害訓練に関わるログとして見做し，省察を行う [7].

UI の実装は CLI によるものとグラフィカルユーザインターフェイス (Graphical User Interface: GUI) によるものを想定し，本稿では後者について説明を行う．災害シナリオ作成と実施のために訓練実施者に特殊な環境の用意を要求させないため，GUI は Web ブラウザで操作できる実装とし，ライブラリとして NodeJS[*19] を用いた．

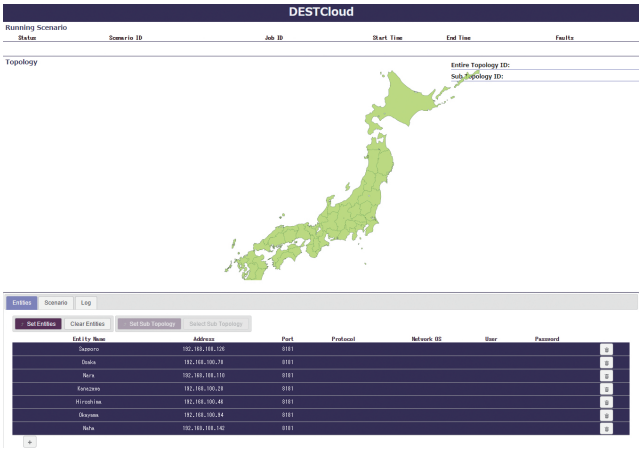

図 6 DESTCloud GUI のネットワーク機器登録画面
Fig. 6 A screen capture of DESTCloud GUI to submit network entities

図 6 は実装した GUI を用いて訓練実施者がネットワーク機器の情報を登録している画面である．訓練実施者は左下にある「+」ボタンをクリックして重畳表示されるダイアログに必要情報を入力する．全てのネットワーク機器を登録し終えたあと「Set Entities」ボタンを押すと entire topology の作成を DSC に要求する．DSC が entire topology を作成すると画面には登録されたネットワーク機器からなるトポロジが表示される (図 7).

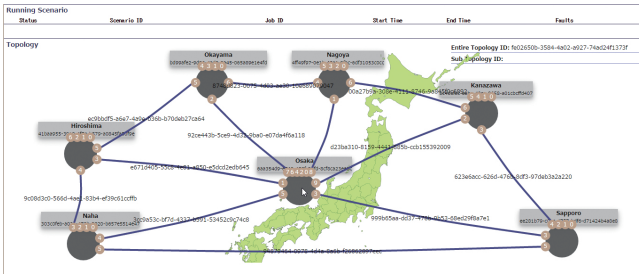

図 7 DESTCloud GUI の全体像表示画面
Fig. 7 A screen capture of DESTCloud GUI displaying entire topology

トポロジはルータが灰色の円で表示され，ルータが保有するインターフェイスがその円の縁に配置された肌色の円で表現される．ルータのインターフェイスと他のルータのインターフェイスが藍色の曲線で結ばれ，これが回線を表現する．ルータ，インターフェイス，回線にはそれぞれ ID が振られており，この ID はそれぞれの表現にマウスのポインタをロールオーバーすることで表示することができる．訓練実施者はこのトポロジの中で訓練に利用する要素からなるトポロジ (sub topology) を選択し，指定する．指定すると，訓練実施者はそれぞれの要素をクリックし，ど

[*19] https://nodejs.org/en/

のような障害表現を発生させるかを選択させることができる．ルータ，インターフェイス，回線によって選択可能な障害表現は異なる．障害を既に与えた要素に対しては，障害を元に戻す「recover」の選択肢が有効になる (図 8)．

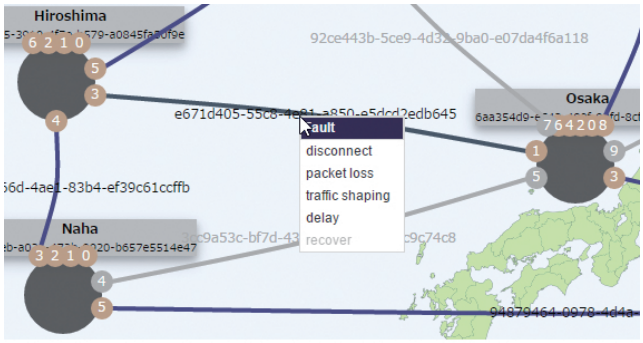

図 8 DESTCloud GUI の障害表現選択画面
Fig. 8 A screen capture of DESTCloud GUI to select network failures

複数の障害表現を登録して訓練シナリオの作成を完了させる．完成した訓練シナリオに対して，訓練実施時間を絶対時間で指定する．訓練シナリオに対しても ID が与えられるため，作成した訓練シナリオは繰り返し実施することができる．また作成した訓練シナリオを呼び出して，修正を施し，異なる ID を付与した訓練シナリオとして登録することができるため，訓練シナリオの再利用も容易である．

3.2 評価

4 拠点を 7 基のルータ，12 の回線からなるネットワークで接続した広域分散ストレージ検証実験を行った．訓練実施者は初歩的な実装による CLI と，今回設計し実装を行った GUI で以下のシナリオの作成を行った．

- シナリオ 1
  - (1) 30 秒後にルータ A が他ルータと接続している全てのインターフェイスをダウンさせる．
  - (2) 300 秒後に全てのインターフェイスを回復させる．
- シナリオ 2
  - (1) 10 秒後にルータ A とルータ B を結ぶインターフェイスをダウンさせる．
  - (2) 11 秒後にルータ A とルータ C を結ぶインターフェイスをダウンさせる．
  - (3) 20 秒後にルータ A とルータ D の回線の遅延時間を 100ms にする．
  - (4) 60 秒後にルータ D とルータ C，ルータ E を結ぶインターフェイスをダウンさせる．
  - (5) 120 秒後に全ての障害を回復させる．

各々のシナリオについて 10 回の作成を行い，シナリオの作成に要した時間の最大値と最小値を除いた 8 つの記録の平均値と分散を表 2 に示す．括弧内が分散であり，平均値，分散ともに単位は秒である．間違ったシナリオ記述になった場合の修正の時間を含める．

表 2 シナリオ記述時間の比較
Table 2 comparisons to describe scenarios

＼	シナリオ 1	シナリオ 2
CUI	168.5 (32.4)	508.4 (108.8)
GUI	28.6 (5.8)	58.8 (6.5)

シナリオ 1 では GUI によるシナリオ作成時間が CUI で要する時間の 17%で済んでおり，シナリオ 2 では 11.5%である．より複雑なシナリオを作成する場合において効率性が高くなると考えられる．また CUI でシナリオを作成した場合，他の作成時間に比べて分散が大きいのは間違ったシナリオを修正する時間が発生しているためである．

4. まとめと今後の課題

広域分散アプリケーションの耐障害性・耐災害性を検証するためのプラットフォームにおける訓練の実施と，訓練に要するコストを低減するための GUI の設計とその評価を行った．設計した GUI は CUI に比べて訓練シナリオを作成する時間を 80%以上低減させることが分かった．今後はシナリオの再利用性をさらに高める機能の具備，結果の可視化の開発を推進する．

参考文献

[1] George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems: Concepts and Design, 5th edition, ISBN: 9780132143011, Addison-Wesley Publishing Company (2011)
[2] 岡村健志, 菊池豊, 福本昌弘, 豊永昌彦, 佐々木正人, 今井一雅, 山田覚, 風間裕, 一色健司, 名和真一, 高畑貴志: 地域 IX における人為的障害による耐災害性の検証, マルチメディア，分散，協調とモバイル (DICOMO2014) シンポジウム, pp.485-489 (2014)
[3] 菊池豊, 岡村健志, 福本昌弘, 豊永昌彦, 佐々木正人, 今井一雅, 山田覚, 風間裕, 一色健司, 名和真一, 高畑貴志, 栢分正人, 井上望美, 柴田祐輔: 地域 IX で恣意的な障害を発生させることによる耐障害性の検証, ITRC technical report 2013 (2014)
[4] Harrington, D., Presuhn, R. and Wijnen, B.: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, RFC3411 (INTERNET STANDARD) (Dec. 2002). Updated by RFCs 5343, 5590.
[5] Enns, R., Bjorklund, M., Schoenwaelder, J. and Bierman, A.: Network Configuration Protocol (NETCONF), RFC 6241 (Proposed Standard) (June 2011).
[6] 北口善明, 柏崎礼生, 近堂徹, 市川昊平, 西内一馬, 中川郁夫, 菊池豊: 広域分散システムの耐障害性を評価する検証プラットフォームの実装と評価, 情報処理学会論文誌, Vol. 57, N.o 3, pp. 958–966 (2016-03-15)
[7] 北口善明, 柏崎礼生, 近堂徹, 市川昊平, 西内一馬, 中川郁夫, 菊池豊: 耐障害性・耐災害性の検証・評価・反映プラットフォームの設計と実装, 研究報告インターネットと運用技術（IOT）, Vol. 2015-IOT-32 (2016)