DESTCloud:

Disaster Emulation and Simulation Testbed for Distributed Computing Environment

Kohei Ichikawa (NAIST),
Hiroki Kashiwazaki (Osaka Univ.),
Yoshiaki Kitaguchi (Kanazawa Univ.),
Tohru Kondo (Hiroshima Univ.),
Kazuma Nishiuchi (Citynet),
Ikuo Nakagawa (Intec Inc., Osaka Univ.),
Yutaka Kikuchi (Kochi Univ. of Technology)

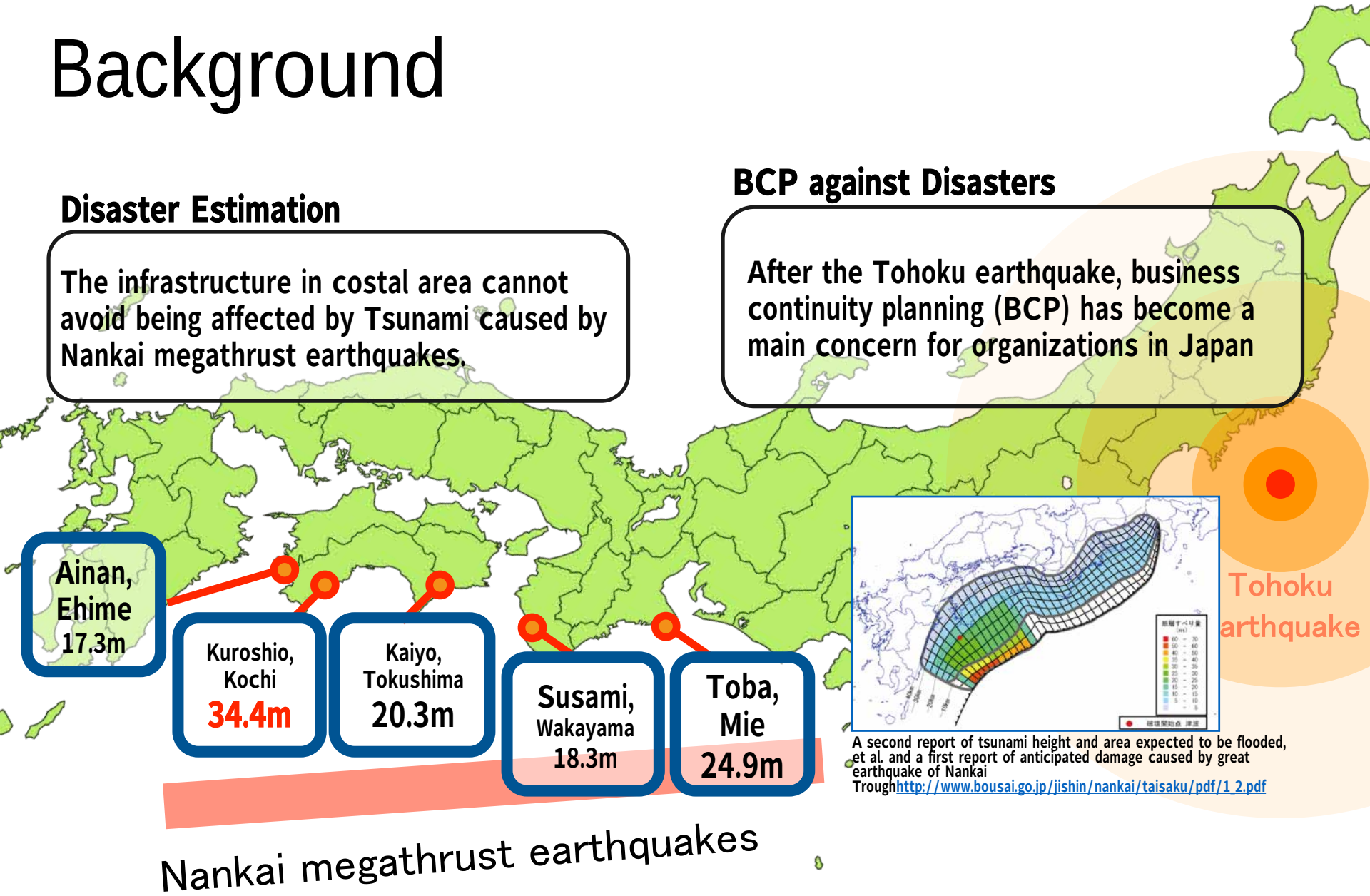
Background

Disaster Estimation

The infrastructure in costal area cannot avoid being affected by Tsunami caused by Nankai megathrust earthquakes.

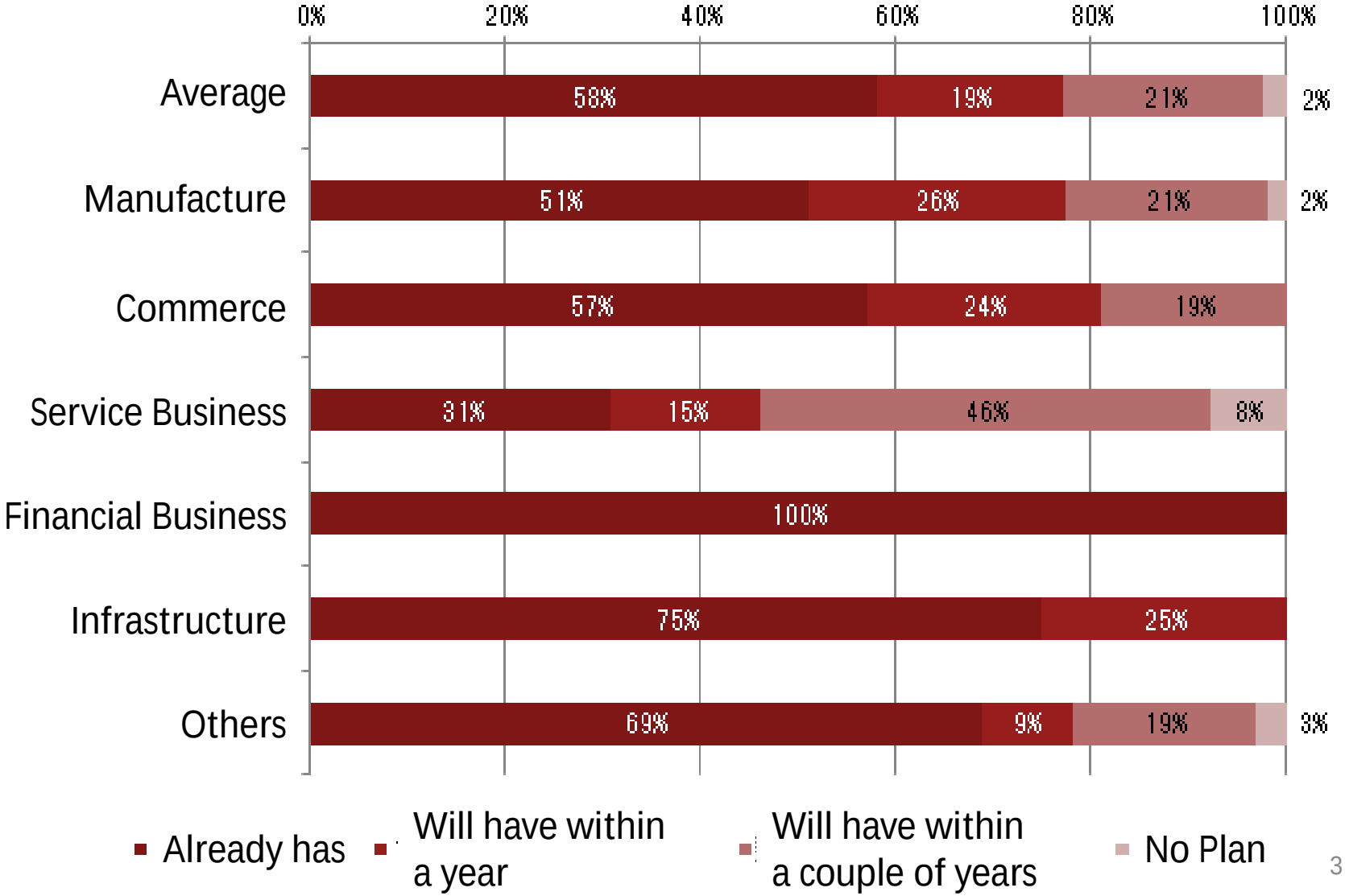
BCP against Disasters

After the Tohoku earthquake, business continuity planning (BCP) has become a main concern for organizations in Japan

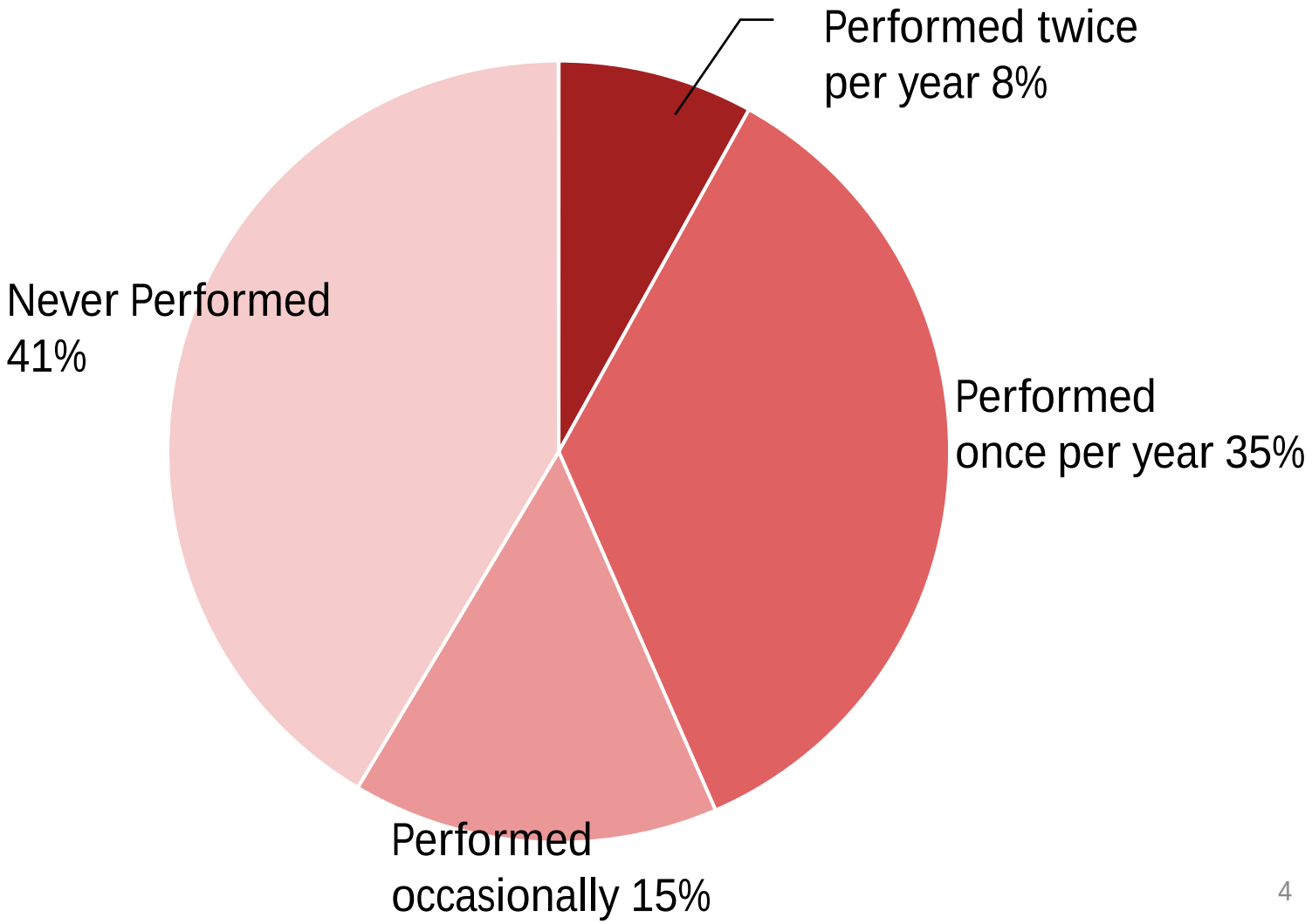


A second report of tsunami height and area expected to be flooded, et al. and a first report of anticipated damage caused by great earthquake of Nankai Trough http://www.bousai.go.jp/jishin/nankai/taisaku/pdf/1_2.pdf

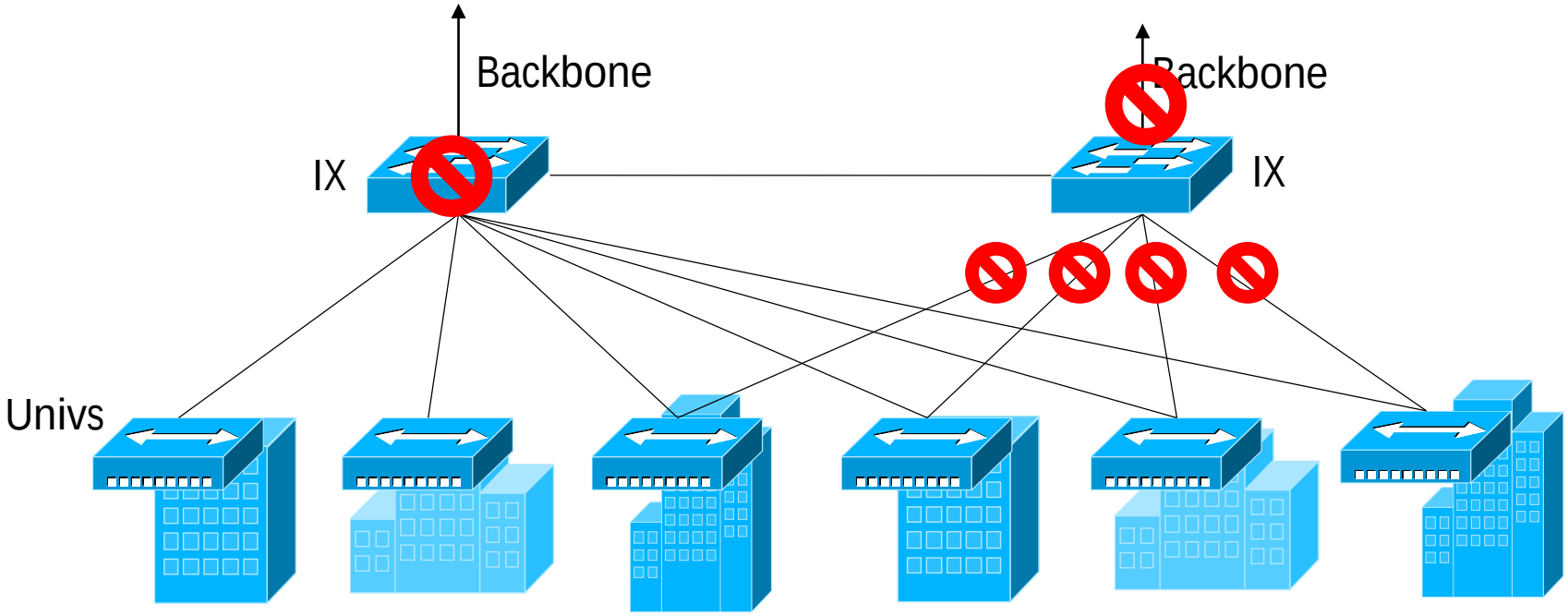
Development of BCP in Japan (survey in 2013)



Evaluation of BCP by Performing Disaster Drills and Practices



Case Study in Kochi Academic Information Network: Disaster Drill for Network Infrastructure



Observed what will happen if network failures occur.

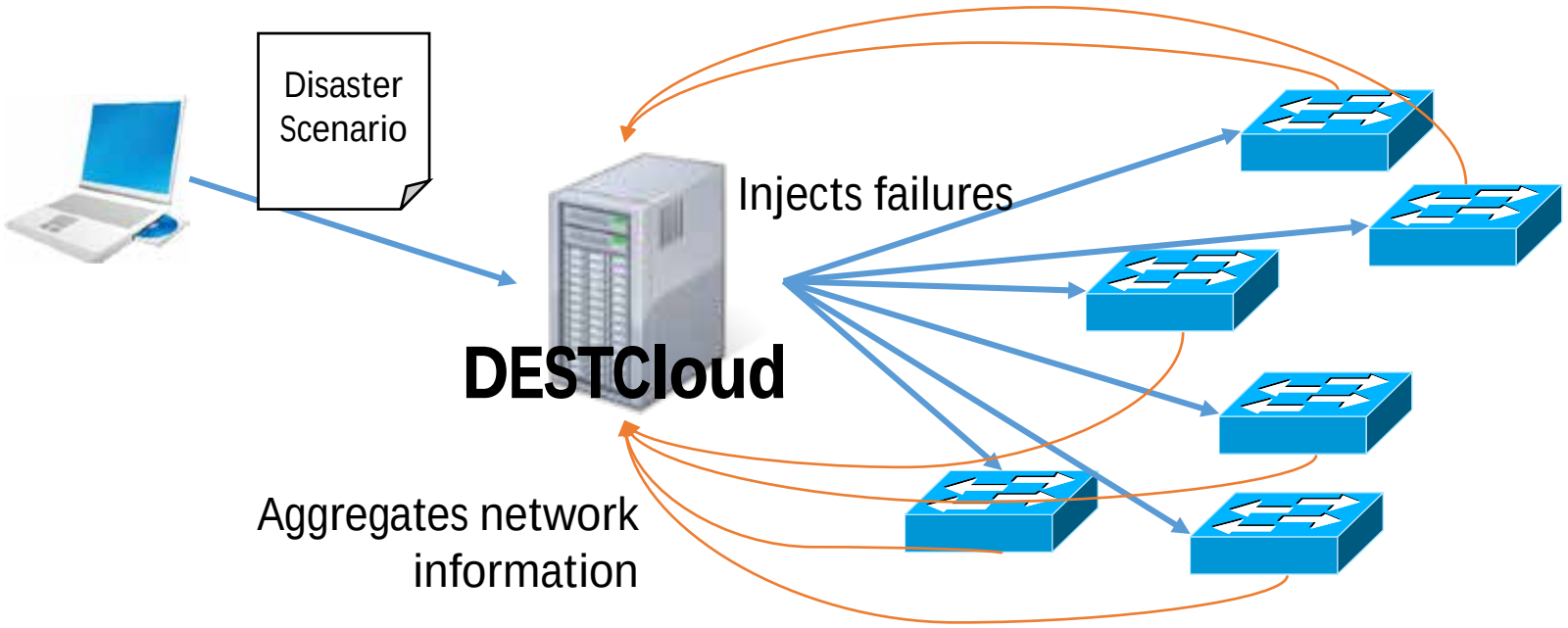
Evaluated & gave feedback to the process of their BCP.

Issues: Manual operations for injecting failures

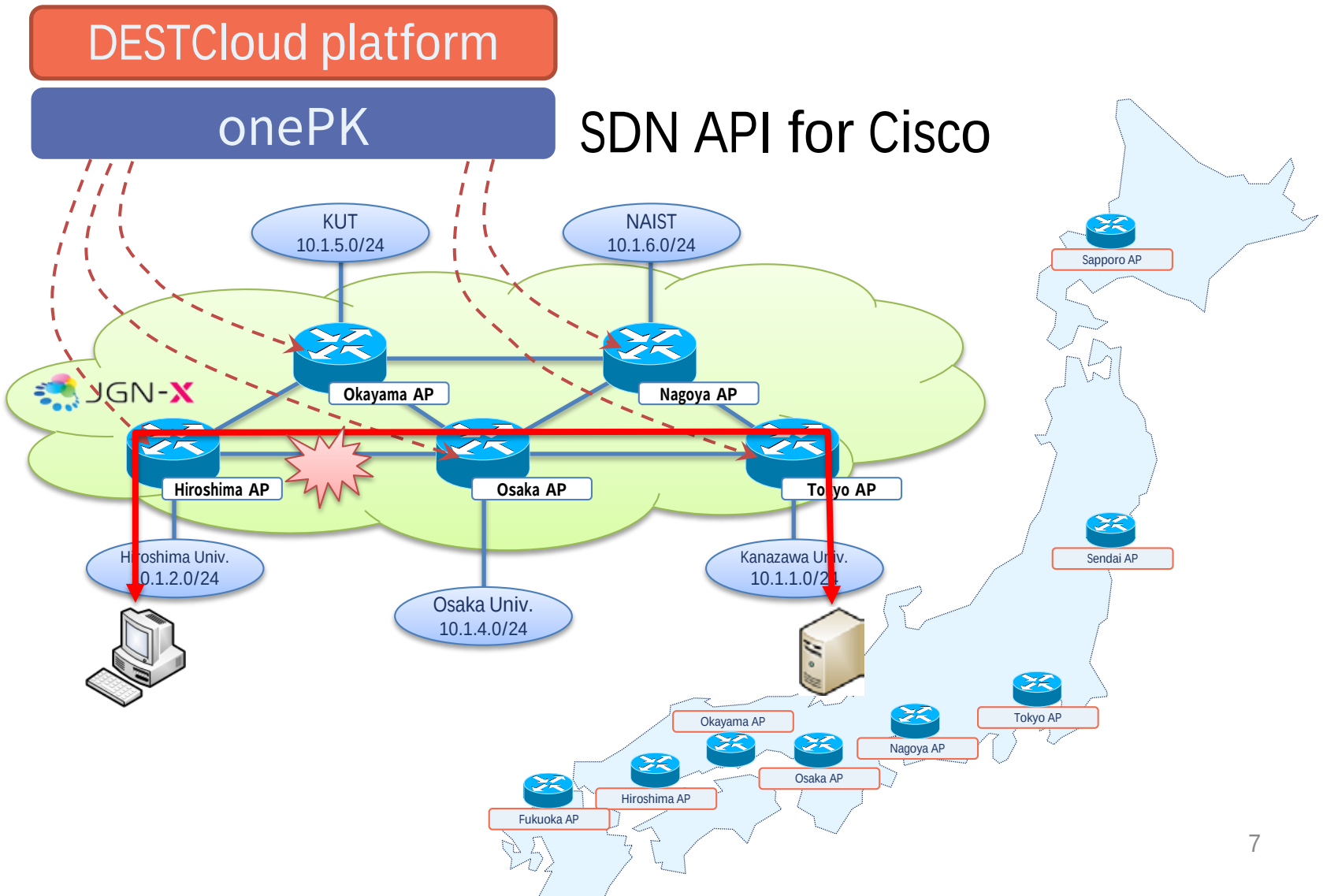
- Only simple disaster testcases
- Repeatability
- Aggregation of Logs

DESTCloud:
SDN for Destroying Cloud

- Emulates disaster scenarios
  - Injects network failures following a disaster scenario
- Performed on actual network environments
- Recovers to original normal status if ends
- Aggregates information and logs during emulations



Testbed Developed on JGN-X with onePK-enabled Switches



onePK

VS

- Basically, a onePK switch behaves as a normal network switch (OSPF, BGP, etc.)
- Gives priority to the configurations (interface, routing, ACL, QoS, etc.) installed via onePK API
- Developer just needs to implement additional functions on existing network

- An OpenFlow switch does not anything without a controller
- Developer needs to design a new network mechanism from scratch

Classification of Network Failures

Cause of failure	Failure factor	Phenomenon	Implementation
Control, operation or software	Traffic regulation control	Congestion	Traffic shaving
	Incorrect route advertisement	Routing loops	Forced update on Routing tables
		Routing flaps	
		Routing failure (unknown destination)	
Network equipment	Equipment failure (entirely)	Communication lost (entirely)	Interface down
	Equipment failure (partly)	Communication lost (partly)	
	Overload of equipment	Packet loss	Traffic shaving
		Increase of latency	Add latency
Communication line	Cable disconnection	Communication lost (partly)	Interface down Traffic shaving
	Failure of repeater/switch		
	Concentration of traffic	Congestion	Traffic shaving
Facility	Building damage	Communication lost (entirely)	Interface down Traffic shaving
	Loss of power supply		
	Failure of air-conditioning	Communication lost (partly)	

The Ministry of Internal Affair in Japan: “an investigative commission of the ways to secure communication methods in emergency such as wide-scale disasters”, “criteria of safety and reliability of information communication network”

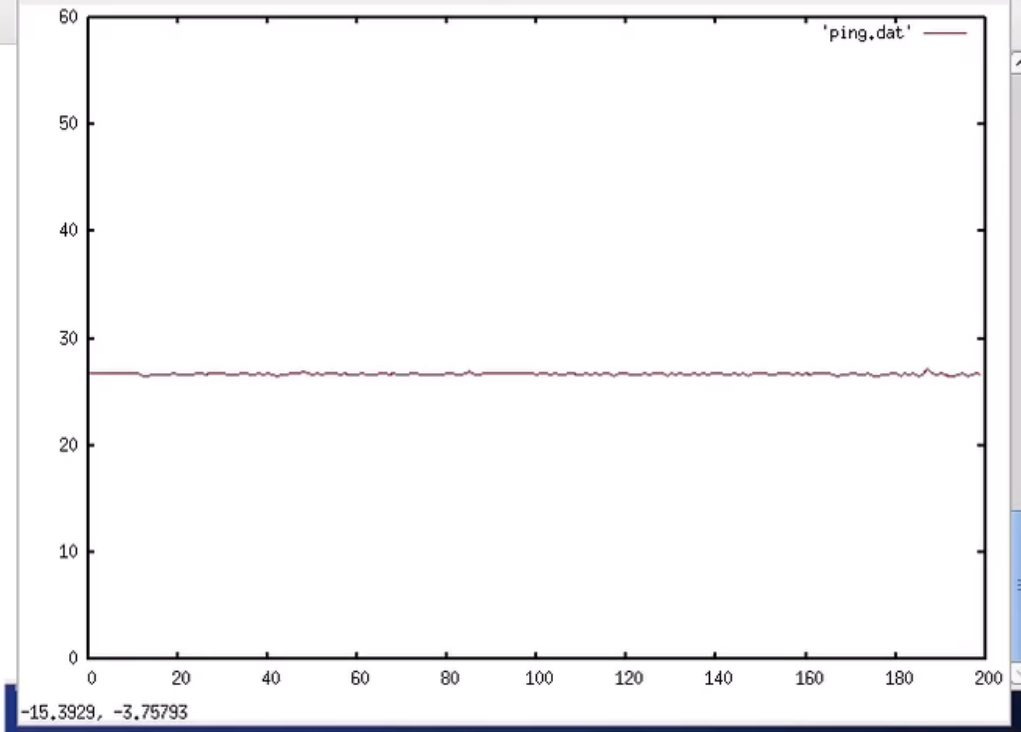
Demo: Link failures in BGP environments

ichikawa@onepk-hu02:~/ping-plot

ファイル(F) 編集(E) 表示(V) 検索 (S) 端末(T) ヘルプ(H)

```
64 bytes from 10.1.1.100: icmp_seq=552 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=553 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=554 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=555 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=556 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=557 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=558 ttl=61 time=26.5 ms
64 bytes from 10.1.1.100: icmp_seq=559 ttl=61 time=27.7 ms
64 bytes from 10.1.1.100: icmp_seq=560 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=561 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=562 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=563 ttl=61 time=26.8 ms
64 bytes from 10.1.1.100: icmp_seq=564 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=565 ttl=61 time=26.5 ms
64 bytes from 10.1.1.100: icmp_seq=566 ttl=61 time=26.5 ms
64 bytes from 10.1.1.100: icmp_seq=567 ttl=61 time=26.5 ms
64 bytes from 10.1.1.100: icmp_seq=568 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=569 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=570 ttl=61 time=26.6 ms
64 bytes from 10.1.1.100: icmp_seq=571 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=572 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=573 ttl=61 time=26.7 ms
64 bytes from 10.1.1.100: icmp_seq=574 ttl=61 time=26.6 ms
```

Gnuplot



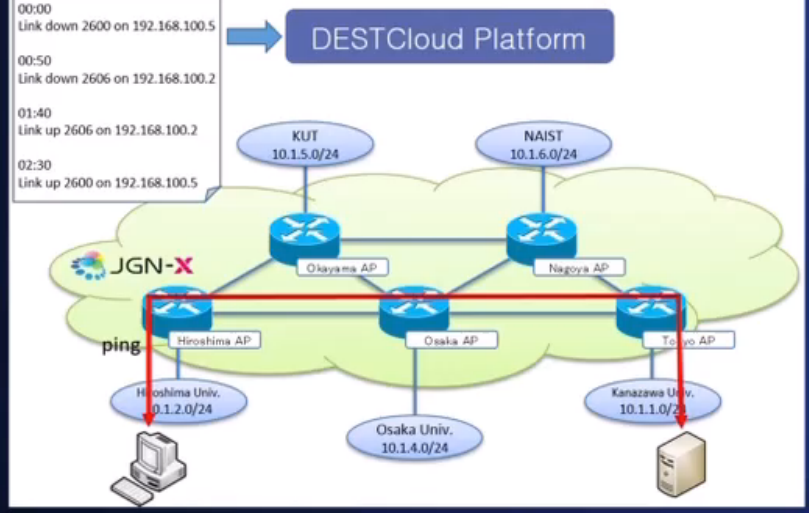
Scenario

00:00
Link down 2600 on 192.168.100.5

00:50
Link down 2606 on 192.168.100.2

01:40
Link up 2606 on 192.168.100.2

02:30
Link up 2600 on 192.168.100.5



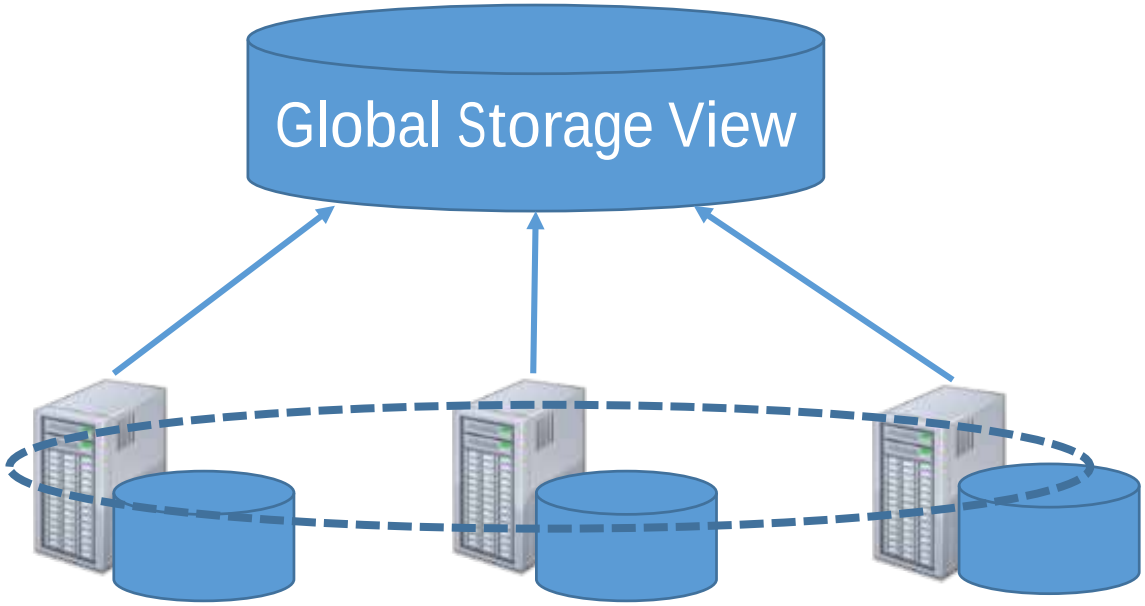
root@ricc-onepk:~/bin/sc

ファイル(F) 編集(E) 表示(V) 検索 (S) 端末(T) ヘルプ(H)

```
[root@ricc-onepk sc]# ./demo.sh
```

Evaluation with actual distributed applications

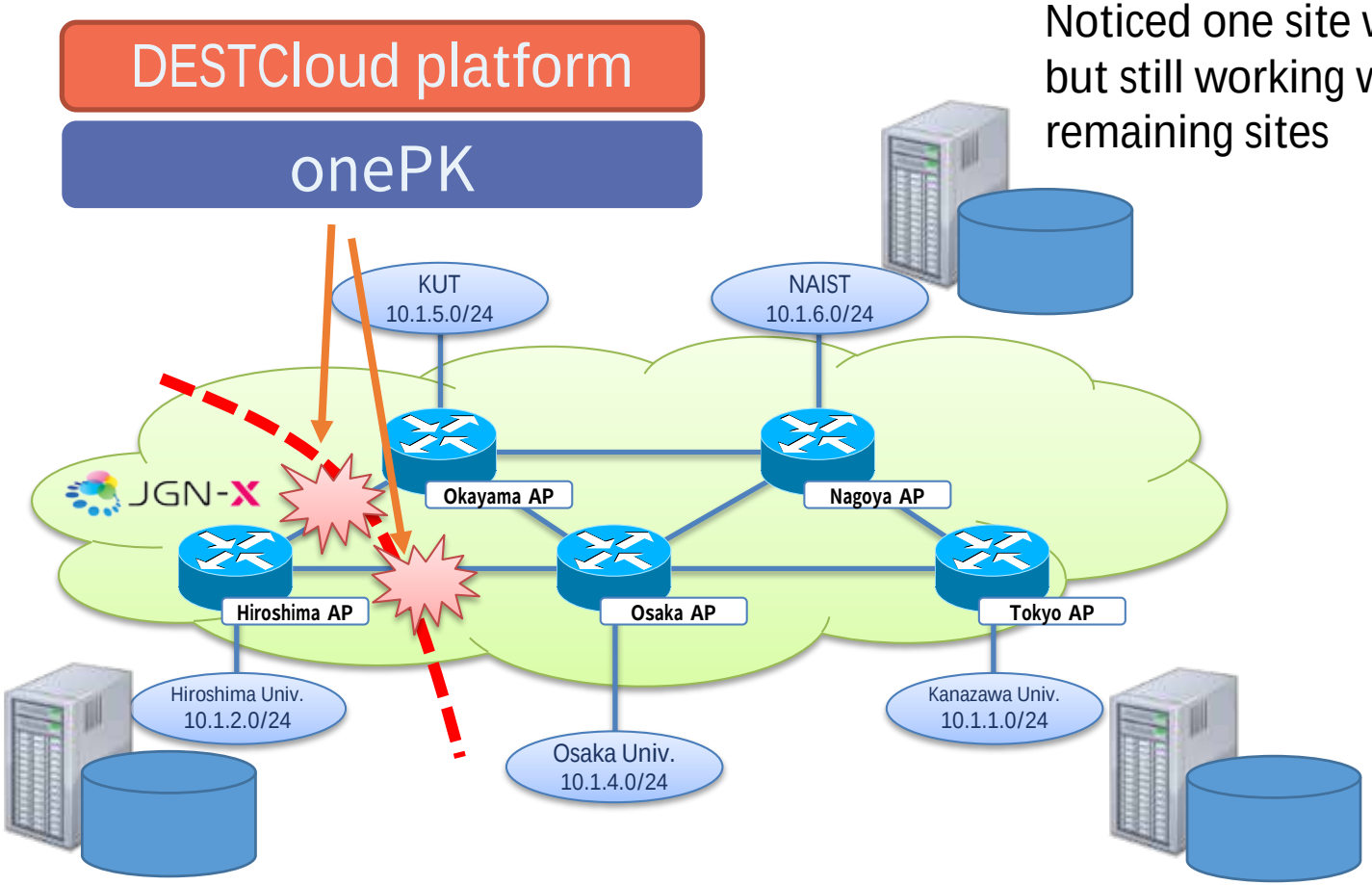
- EXAGE (Wide-area distributed storage)



I/O operations are distributed based on a P2P mechanism.
Consistency of data is the main concern of this application.

Needs a lot of evaluations on various situations

Evaluation of Sprit-Brain Problem



Noticed one site was down, but still working with the remaining sites

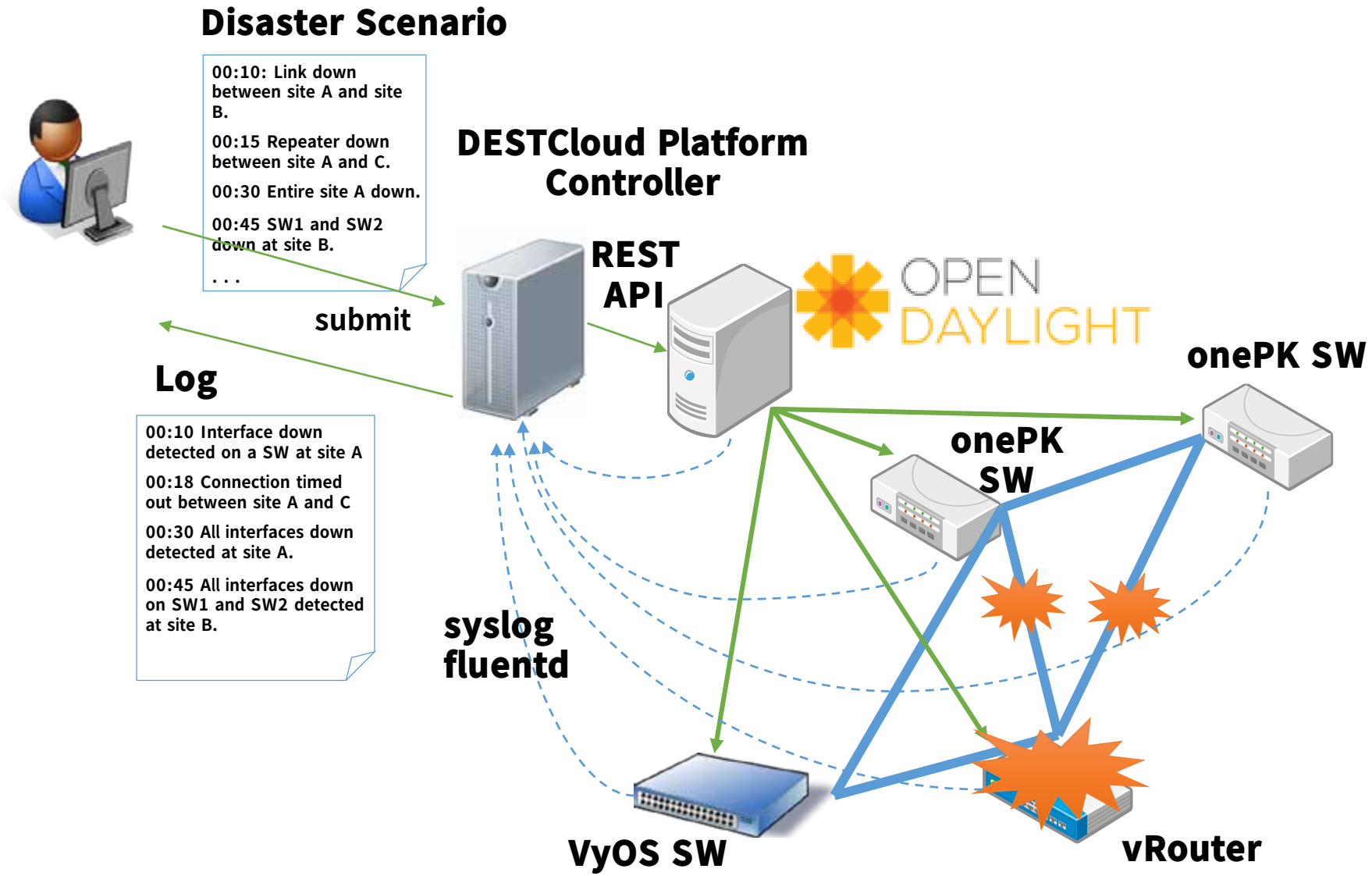
Noticed this site was isolated from the others and suspended

Noticed one site was down, but still working with the remaining sites

Results

- The I/O performance was degraded during failures.
- Consistency was still kept after the failures are recovered.
- But, we found a file was broken sometimes if we kept reading the file during the failures.
  - This is actually caused by a bug of EXAGE
  - Our platform helped to fix it.

New Design


Disaster Scenario
00:10: Link down between site A and site B.
00:15 Repeater down between site A and C.
00:30 Entire site A down.
00:45 SW1 and SW2 down at site B.
. . .
submit
Log
00:10 Interface down detected on a SW at site A
00:18 Connection timed out between site A and C
00:30 All interfaces down detected at site A.
00:45 All interfaces down on SW1 and SW2 detected at site B.
DESTCloud Platform Controller
REST API
OPEN DAYLIGHT
onePK SW
onePK SW
syslog
fluentd
VyOS SW
vRouter

Conclusion

- Developed DESTCloud, a disaster emulation platform for distributed systems
  - Implemented with onePK technologies
  - Injects various network failures on an actual network environment by following a disaster scenario
  - Helps to evaluate distributed systems on an actual network environment

Future Plan

- Design based on more common standards
  - Open Daylight architecture
  - REST API

Project Members

gzr gtlo gpvcr"eqpuqtvkwo
d{"wpkxgtukvkgu"tgugctej "gpvkvkgu"gve0

J ltquj lo c Wpkx0

Lcr cp"Cf xcpegf Kpuvkwvg"qh Uelgpeg"cpf "Vgej pqmqi {

Mtpc/cy c Wpkx0

Mqvq Wpkx0

WE "Ucp"Fkgi q
UC San Diego

Mwuj w Wpkx0

Qucmc Wpkx0

Pcvqpcn"Kpuvkwvg qh"Kphqto cvkqp"cpf " Eqo o wplecvkqp"Vgej pqmqi {

Kpvge."Kpe0
INTEC
IT Holdings Group

Mtej k"Kpuvkwvg qh"Vgej pqmqi {

Pctc"Kpuvkwvg"qh Uelgpeg"cpf "Vgej pqmqi {

Mwuj w"Kpuvkwvg qh"Vgej pqmqi {

Pcvqpcn"Kpuvkwvg"qh" Kphqto cvkegu

NII
Notional Institute of Informatics

TKEE "Rtqlgev"Ogo dgtu