

Dockerコンテナを用いたLayer2演習に対応可能な IPネットワーク構築演習支援システムの開発



◎ 菅家悠希[†], 井口信和^{†‡}
[†]近畿大学 理工学部 情報学科
[‡]近畿大学 情報学研究所

目次

- 研究概要

 - 背景

 - 既存システムの課題

 - 目的・アプローチ

- 研究内容

- 実験

- まとめ

研究背景

■ ネットワークの需要増加[1]

年月	2018年5月	2019年5月	2020年5月	2020年11月
帯域(Tbps)	10.2	12.0	19.0	19.8

■ IT技術者の不足[2]

年	2020年	2025年	2030年
不足数(万人)	30	36	45

ネットワーク人材の養成が急務

[1] 総務省:令和3年度情報通信白書,
<<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nb000000.html>>
(参照:2022-01-03)

[2] 経済産業省: I T 人材需給に関する調査 (概要) ,
<https://www.meti.go.jp/policy/it_policy/jinzai/gaiyou.pdf> (参照:2022-01-03)

研究背景

■ ネットワーク技術者の養成

□ IPネットワーク構築演習

- ルータやスイッチなどの実機を操作して演習実施

□ CNA (Cisco Networking Academy) [3]

- ネットワーク構築演習
- ネットワークトラブルシューティング演習

■ 課題

□ ルータやスイッチが高価

□ 物理的スペースの確保

□ コロナ過の中対面での実施が不可欠

既存システム(NPL: NetPowerLab) [4]

- IPネットワーク構築演習支援システム
 - UML (User Mode Linux) [5]を使用
- 仮想ネットワーク機器を用いた演習が可能
 - 実機を使う必要がない
 - 物理的スペース問題も解決
- Webブラウザ上で動作
 - 場所にとらわれずに演習が可能

[4] Nobukazu Iguchi: Development of a self-study and testing function for NetPowerLab, an IP networking practicesystem, International Journal of SpaceBased and Situated Computing, Vol. 4, No. 175-183 (2014).

[5] User Mode Linux, 入手先<<http://user-mode-linux.sourceforge.net/>>, (参照:2022-01-03) .

NPLの課題

- UMLを用いた課題
 - ドキュメントの不足
 - UMLの開発が終了
 - UMLの起動時間が長い
- Layer2プロトコルの演習に対応していない
 - VLAN, STP, EtherChannel等の演習ができない

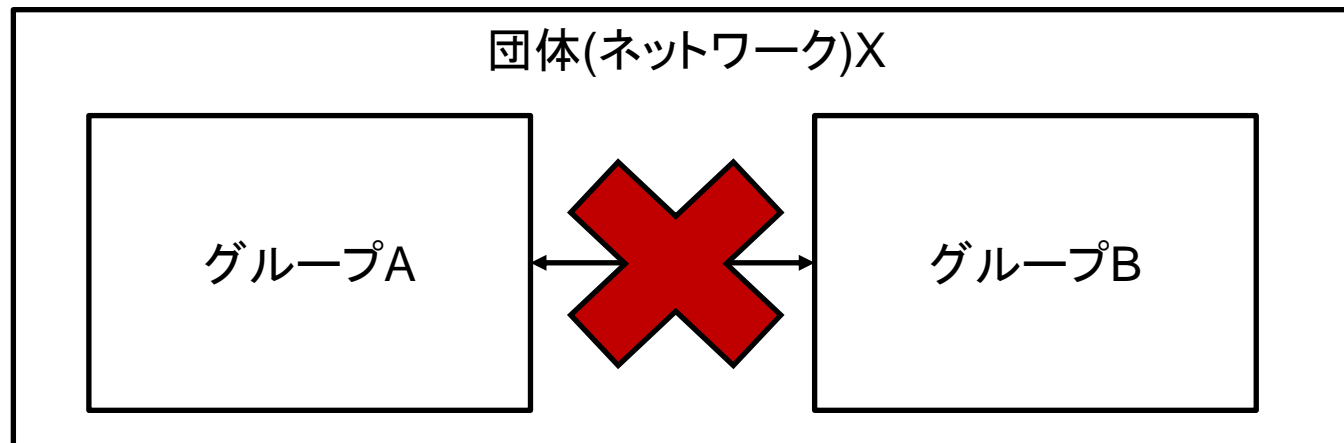
補足: Layer2について

■ Layer2

- OSI参照モデルの第二層(データリンク層)
- 直接接続された機器とのルールを定める

■ Layer2技術の例: VLAN

- 論理的にネットワークをグループに分割する技術
- グループ内の通信を許可しその他は禁止する
といった用途に使用



目的・アプローチ

■ 目的

- 学習者のネットワーク構築学習環境の利便性向上

■ アプローチ

- 開発環境・ネットワークの起動速度の改善
 - 使用技術の見直し
 - システムアーキテクチャの一新
- Layer2 Switchの追加により演習の幅を拡張

目次

- 研究背景
- 研究内容
 - システムアーキテクチャの見直し
 - 対応技術
 - システム構成
 - コマンド変換機能
 - ネットワーク保存機能
 - 課題演習機能
- 実験
- まとめ

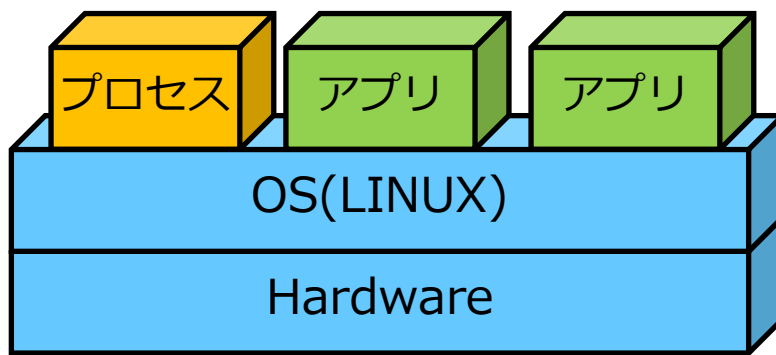
システムアーキテクチャの見直し

■ UML

- UML上でソフトウェアやOSを動作させることで仮想ネットワーク機器を実現可能

■ Docker[6]

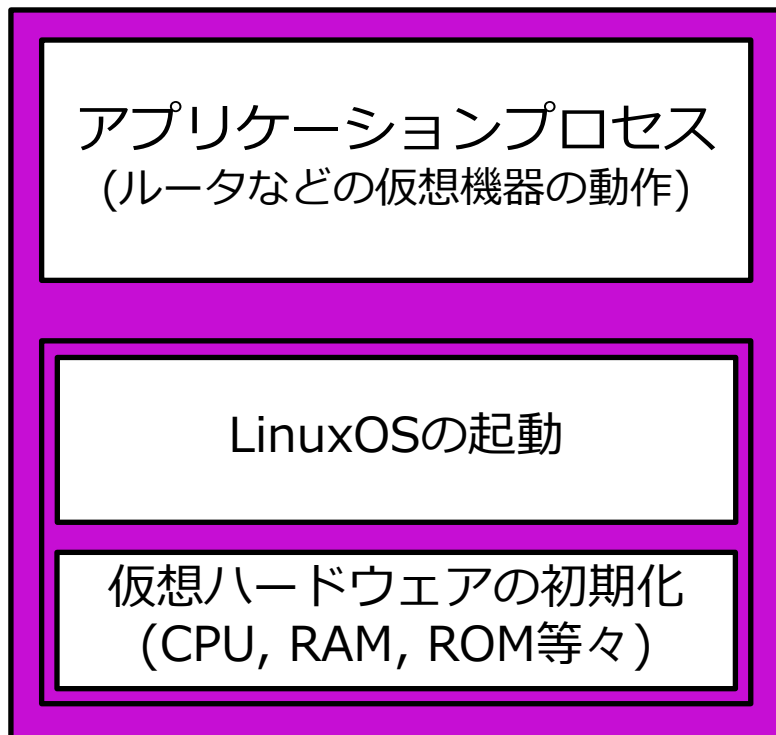
- コンテナ上でソフトウェアやOSを動作させることで仮想ネットワーク機器を実現可能
- 迅速にコンテナを起動可能



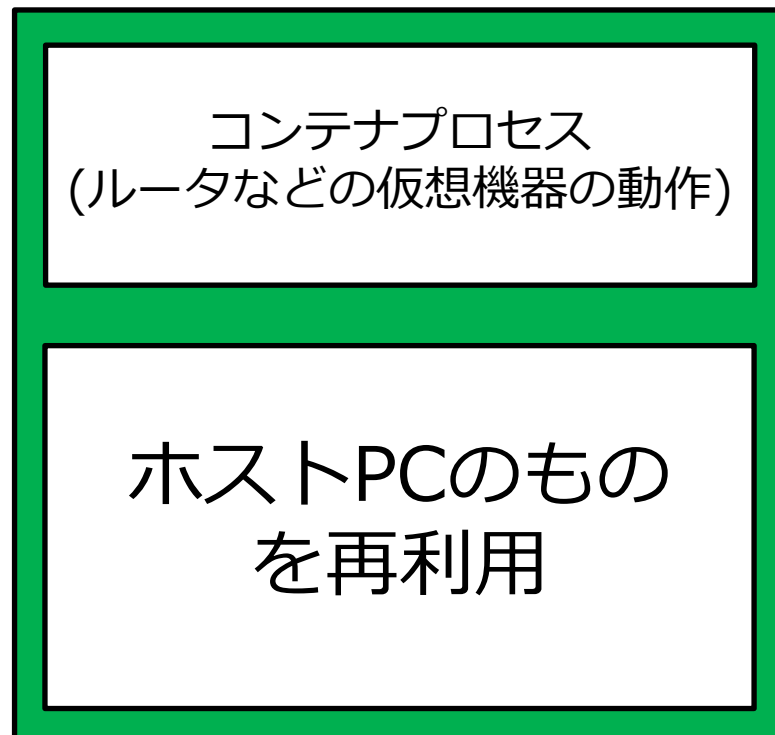
システムアーキテクチャの見直し

■ 各プロセスの比較

UML(仮想マシン)



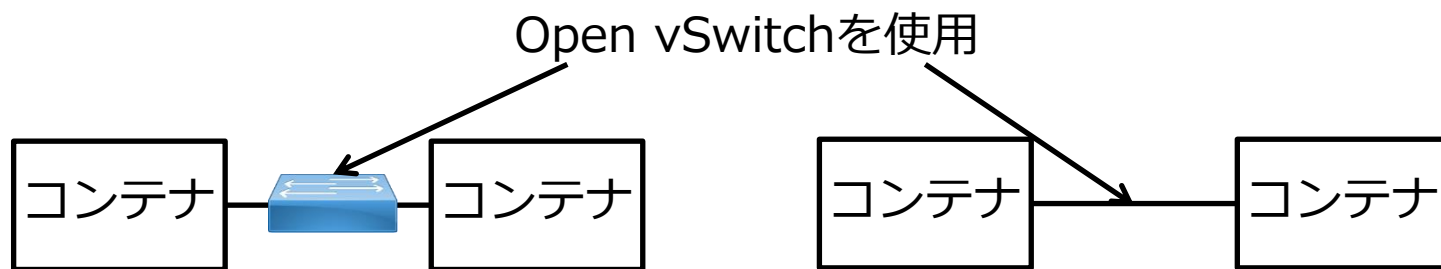
Dockerコンテナ



仮想マシンをブート(起動)
するのがボトルネックに

システムアーキテクチャの見直し

- VyOS[7]
 - 仮想ルータ実現のために使用
- CentOS[8]
 - 仮想ホスト実現のために使用
- Open vSwitch[9]
 - 仮想スイッチ実現及びコンテナ間の結線に使用



[7] VyOS, 入手先 <<https://vyos.io/>>, (参照:2022-01-03) .

[8] The CentOS Project, 入手先 <<https://www.centos.org/>>, (参照:2022-01-03)

[9] Open vSwitch-Overview,

入手先 <<https://www.openvswitch.org/features>>, (参照:2022-01-03)

実装済みの技術について

■ ルータを用いた技術[10]

- 既存システム: Quagga[11] を用いて実装
本システム: VyOS を用いて実装

技術	既存システム(Quagga)	本システム(VyOS)
静的ルーティング	○	○
動的ルーティング(RIP, OSPF)	○	○
冗長化 (パッシブインタフェース)	○	○
ACL	○	△(実装中)
VRRP	×	○
NAT	×	△(実装中)
DHCP	×	△(実装中)

[10] Quagga Routers comp, 入手先 <https://openmaniak.com/quagga_func.php>, (参照:2022-02-08)

[11] Quagga Software Routing Suite, 入手先 <<http://www.nongnu.org/quagga/>>, ¹³ (参照:2022-02-08)

実装済みの技術について

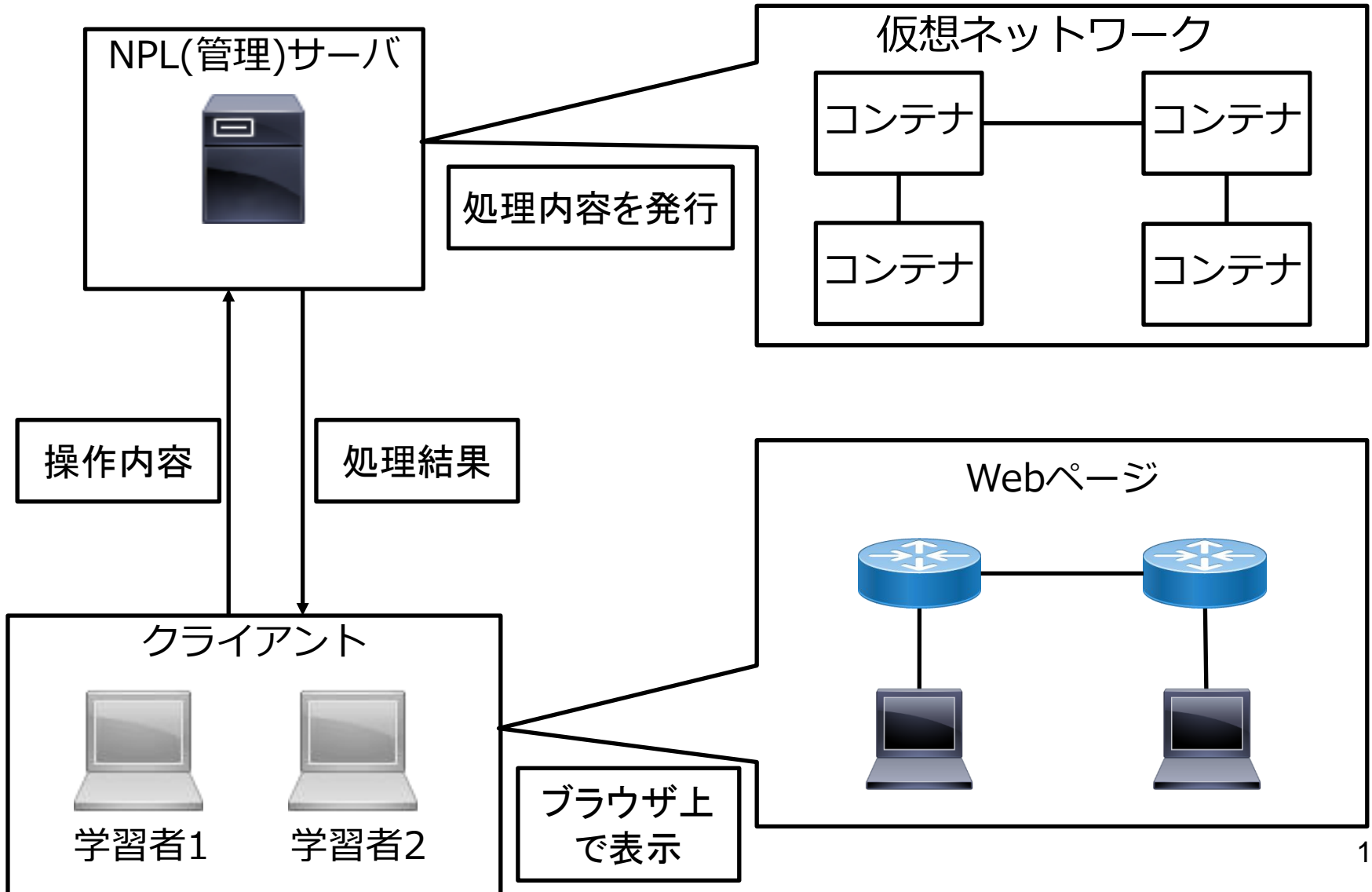
■ スイッチを用いた技術

□ 既存システム: 未実装

本システム: Open vSwitchを用いて実装

技術	既存システム	本システム(Open vSwitch)
VLAN	×	○
EtherChannel	×	○
STP	×	○

システム構成



アプリケーション画面

Device Add

Host

Router

Switch

Hub

Save

Load

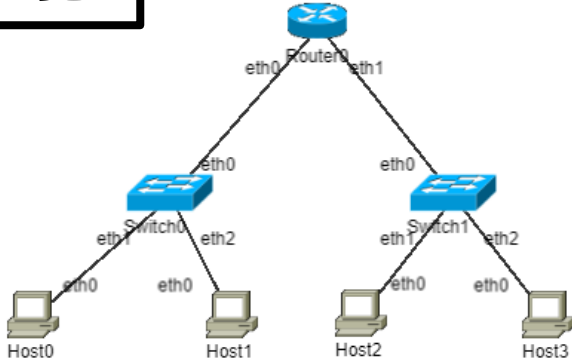
Anime: Off

Animation Config

Network Topology | 結線 結線OFF

デバイス一覧

ネットワークキャンバス



Console | 選択: Router0

Hello, This is Authorized Access Only!
Router>
Router>

コンソール

機器情報一覧

NodeName	Interface	IPAddress	NetMask	MacAddress	Status
Router0	eth0			56:4e:53:a5:b9:51	-----
Router0	eth1			56:4e:53:22:6b:82	-----
Switch0	eth0			56:4e:53:ca:75:77	-----
Switch0	eth1			56:4e:53:f7:9d:3d	-----
Switch0	eth2			56:4e:53:6a:e4:f1	-----
Switch1	eth0			56:4e:53:f6:f7:19	-----
Switch1	eth1			56:4e:53:a7:8e:1c	-----
Switch1	eth2			56:4e:53:48:a9:64	-----

機器情報一覧

コマンド変換機能の実装

■ コマンド変換機能

- 学習者がCisco IOSコマンドの文法に則り入力したコマンドを仮想機器へのコマンドに変換する機能

```
interface ethernet 0  
ip address 192.168.1.1 255.255.255.0
```

Ciscoルータ



```
set interfaces ethernet eth0 address 192.168.1.1/24
```

VyOS



コマンド変換機能の実装

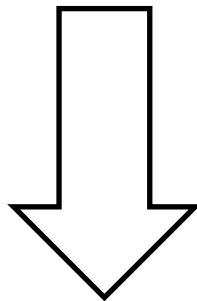
■ コマンド変換機能

- 学習者がCisco IOSコマンドの文法に則り入力したコマンドを仮想機器へのコマンドに変換する機能

変換前のコマンド(CISCO IOS用コマンド)

```
interface ethernet 0
```

```
ip address 192.168.1.1 255.255.255.0
```



変換後のコマンド(VyOS用コマンド)

```
set interfaces ethernet eth0 address 192.168.1.1/24
```

コマンド変換機能の実装

```
Hello, This is Authorized Access Only!  
Router> enable  
{Router, message, user, 01}  
Router# configure terminal  
{Router, message, privilege, 07}  
Router(config)# interface ethernet 0  
{Router, message, global-config, 11, 0}  
Router(config-if)# ip address 192.168.1.1 255.255.255.0  
{Router, message, interface-config, 07, eth0, 192.168.1.1, 255.255.255.0}
```



コマンドid: 07
パラメータ1: eth0, パラメータ2: 192.168.1.1
パラメータ3: 255.255.255.0

変換後のコマンド(VyOS用コマンド)
set interfaces ethernet eth0 address 192.168.1.1/24

ネットワーク保存機能の実装

■ ネットワーク保存機能

- 作業中のネットワークをXMLファイルに保存し作業再開時に復元することが可能

The screenshot displays a network simulation environment. On the left, a 'Device Add' sidebar contains icons for Host, Router, Switch, and Hub. Below this is a 'Tool' section with 'Save' and 'Load' buttons; the 'Load' button is highlighted with a red box and a red arrow. The main area shows a 'Network Topology' with two switches, Switch0 and Switch1, connected by their eth1 interfaces. Switch0's eth0 is connected to Host0, and Switch1's eth0 is connected to Host1. A 'Console' window on the right shows the configuration for Switch1, including 'configure terminal', 'interface ethernet 0', 'switchport mode access', and 'switchport access vlan 20'. A 'Save' dialog box is open in the center, prompting the user to enter a filename, with 'VLAN' entered in the text field. At the bottom, a '機器情報一覧' (Device Information List) table provides details for all devices in the topology.

NodeName	Interface	IPAddress	NetMask	MacAddress	Status
Switch0	eth0			56:4e:53:ba:bf:c6	-----
Switch0	eth1			56:4e:53:14:45:da	-----
Switch1	eth0			56:4e:53:78:50:a6	-----
Switch1	eth1			56:4e:53:33:e20	-----
Host0	eth0	192.168.1.1	255.255.255.0	56:4e:53:89:26:38	-----
Host1	eth0	192.168.1.2	255.255.255.0	56:4e:53:15:c7:e5	-----

ネットワーク保存機能の実装: XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
....
```

```
....
```

```
<line ifdst="Host0" ifdst_port="eth0" ifsrc="Switch0" ifsrc_port="eth0"></line>
```

```
....
```

```
....
```

```
<vm name="Host0" position_x="215" position_y="319">
```

```
  <host name="Host0">
```

```
    <if mac="56:4e:53:6a:ab:a8" name="eth0" status="-----">
```

```
      <ip address="192.168.1.1" netmask="255.255.255.0"></ip>
```

```
    </if>
```

```
  </host>
```

```
</vm>
```

```
....
```

```
....
```

ネットワーク保存機能の実装: デモ

Device Add

Host

Router

Switch

Hub

Tool

Save

Load

Anime: Off

Animation Config

Network Topology | 結線 結線OFF

```
graph TD
    Router0 ---|eth0| Switch0
    Router0 ---|eth2| Switch0
    Switch0 ---|eth0| Host0
    Switch0 ---|eth1| Host1
```

Console | 選択: Host0

```
ping -c 4 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data:
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.258 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.040 ms
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 58ms
rtt min/avg/max/mdev = 0.040/0.096/0.258/0.093 ms
ping -c 4 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data:
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.191 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=0.037 ms
--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 79ms
rtt min/avg/max/mdev = 0.037/0.079/0.191/0.064 ms
```

機器情報一覧

NodeName	Interface	IPAddress	NetMask	MacAddress	Status
Router0	eth0	192.168.1.10	255.255.255.0	56:4e:53:27:8e:16	up
Switch0	eth0			56:4e:53:85:d:e6	_____
Switch0	eth1			56:4e:53:f5:f3:9b	_____
Switch0	eth2			56:4e:53:40:6fa6	_____
Host0	eth0	192.168.1.1	255.255.255.0	56:4e:53:6a:aba8	_____
Host1	eth0	192.168.1.2	255.255.255.0	56:4e:53:21:6f:c8	_____

課題演習機能の実装

■ 課題演習機能

□ 指定されたネットワークを正しく作成できるか 正誤判定可能

Tool

Save

Load

Anime: Off

Animation Config

LogSend

Network Topology | 結線 結線OFF

Console | 選択: Switch0

Hello, This is Authorized Access
Switch>
Switch>
Switch> enable
Switch#
Switch#
Switch(config)# configure terminal
Switch(config)#
Switch(config)# interface ethernet0
Switch(config-if)#
Switch(config-if)# switchport mode access
Switch(config-if)#
Switch(config-if)# % Invalid input detected at end of command.
Switch(config-if)# switchport access vlan 10
Switch(config-if)#
Switch(config-if)# switchport access vlan 10
Switch(config-if)#
Switch(config-if)# switchport access vlan 10
Switch(config-if)#

課題を選択してください

課題番号	進捗	課題名	課題
1	---	Configure VLANs and Trunking	S2H2 VLAN演習

演習内容

別のVLAN上に存在するHost0とHost1同士の疎通について学習しよう。VLANの設定をした後にトランクリンクの設定をしよう。

---アドレステーブル---

Host0 eth0 192.168.1.1/24
Host1 eth0 192.168.1.2/24

Switch0 eth0 -----
eth1 -----
Switch1 eth0 -----
eth1 -----

---STEP1---
アドレステーブルに則って各ネットワーク機器にIPアドレスを設定しよう。

---STEP2---
Hostとの接続を実施しているSwitchの各ポートにVLANを設定しよう。

---STEP3---
Switch同士を接続している各ポートにトランクリンクを設定しよう。

・許可するVLAN: 1,10,20
---STEP4---
・Host0からHost1への疎通をpingコマンドで確認しよう。
・Host1からHost0への疎通をpingコマンドで確認しよう。

課題を設定しています

機器情報一覧 | 演習情報 | 採点情報

採点結果 (✓正解 ✗不正解 ■不要項目)

✗ Switch0
✓ eth0
✓ Status: -----
✓ SWITCHPORT MODE: Access_Port
✓ VLAN ID: 10
✗ eth1
✓ Status: -----
✗ SWITCHPORT MODE: Not Configured
✗ ALLOWED VLAN: Not Configured
✗ Switch1
✗ eth0

採点情報

経過時間: 00:02:33
問題正解数: 12 / 18
正解率: 66%
採点回数: 4回

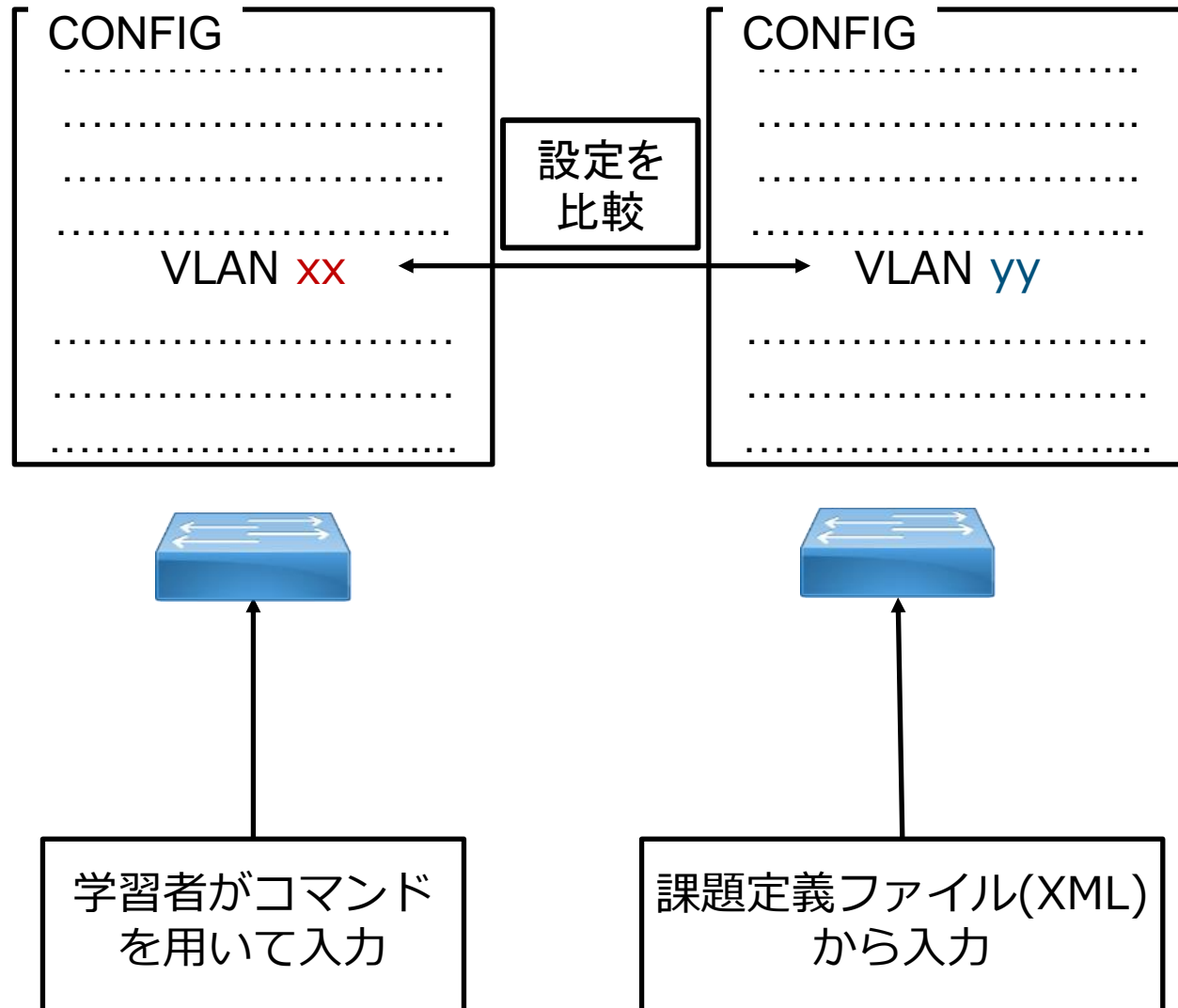
Auto Scoring

動作確認

ping 192.168.1.2
ping 192.168.1.1

Action Check

課題演習機能の実装: 採点方法について



目次

- 研究背景
- 研究内容
- 実験
 - 性能評価実験
- まとめ

性能評価実験

■ 目的

- 本システムと既存システムの仮想機器の起動速度を比較し優位性を確認

■ 実験の方法

- 各ネットワーク機器を50台ずつ起動し50台の起動速度の平均時間・標準偏差を算出
- 以下の環境にてVirtualBoxを用いて実施

ハードウェア	
CPU	CPU: Intel® Core™ i5-10300H CPU @ 2.50GHz 2コア割り当て
RAM	8.0GB 割り当て

性能評価実験

■ 実験結果

機器名	既存システム(s)		本システム(s)	
	平均	標準偏差	平均	標準偏差
Router	9.0	0.7	0.2	0.02
Host	10.7	0.6	0.2	0.03
Switch	---	---	0.02	0.01

起動時間を約1/50程度に
削減することが可能

目次

- 研究背景
- 研究内容
- 実験
- まとめ

まとめ

■ 研究概要

- 背景・既存システムの課題
- 目的・アプローチ

■ 研究内容

- システムアーキテクチャの見直し
- 対応技術・システム構成
- 各種機能

■ 実験

- 性能評価実験

■ 今後の課題

- 演習の幅の拡張
- 利用評価実験の実施

