FPGAを使ってOpen vSwitchのデータプレーンを作る

慶應義塾大学　空閑洋平, 松谷健史
<sora@haeena.net>

SDN Japan 2012/12/7

概要

- Open vSwitchを使ったDIY設計スイッチの発表です
  - 5000円前後のFPGAを想定
  - 1000BASE-T マルチポート
  - NIC Offloading機能
  - CPU+SW部と転送HW部の分離
  - Cut-through forwarding

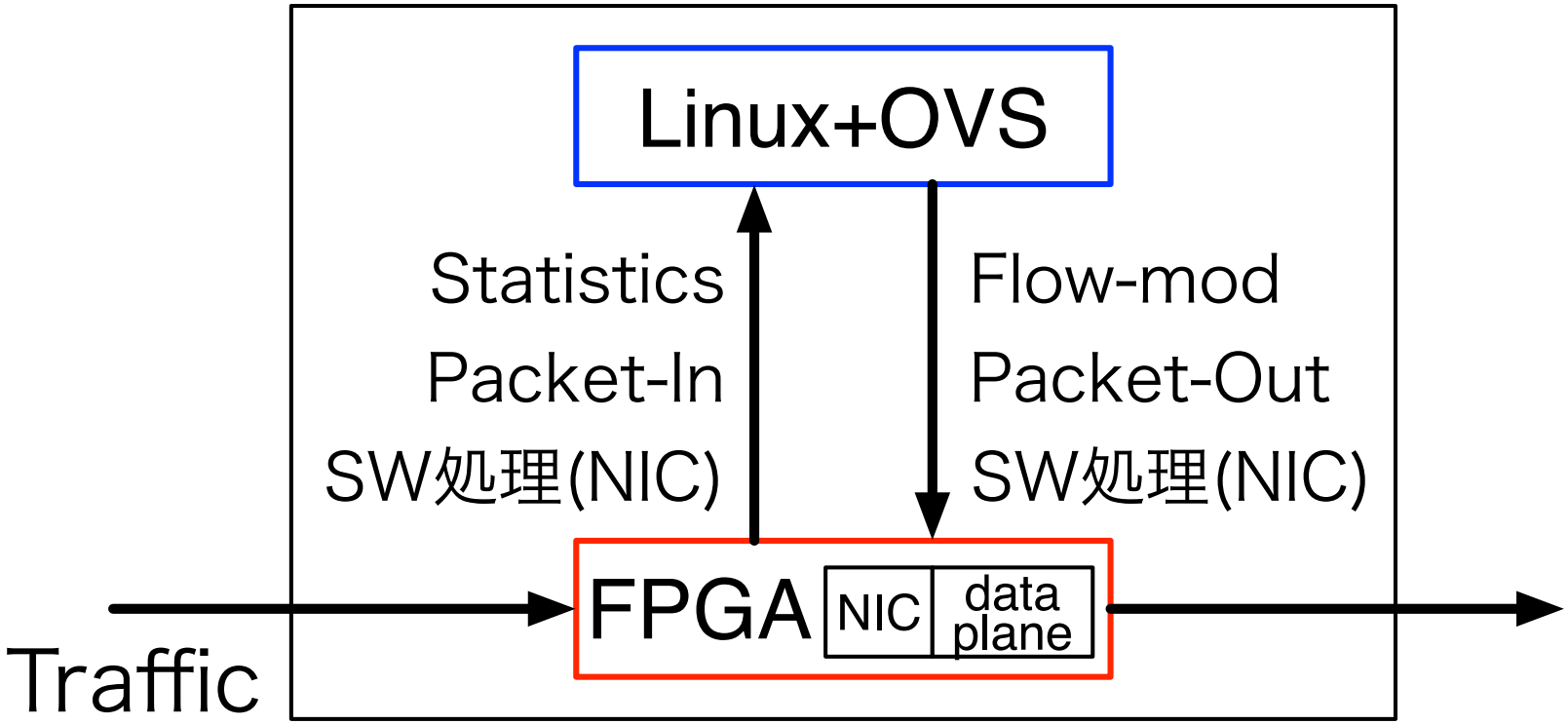
- 筐体作成の検討
  - 2万円前後で作るOpen vSwitch筐体

OpenFlowとスイッチ開発環境

- NetFPGA
  - 1Gと10Gの4ポート
  - 最近10G向けのOpenFlow実装が公開
- Broadcom API
  - Indigo firmwareやPica8，など
- OpenWRT
  - ソフトウェアで処理
- Open vSwitch
  - HW化しやすいように設計
- OFTest
  - スイッチの機能テスト

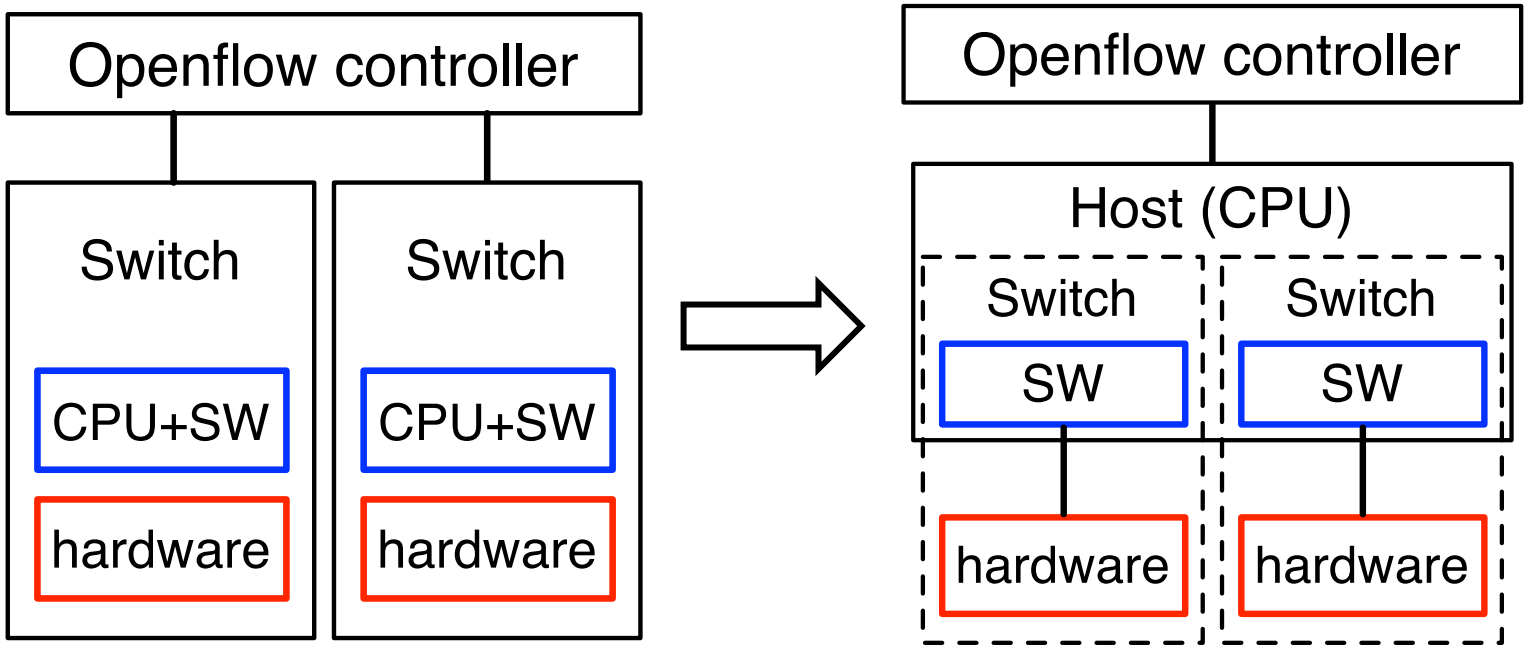
スイッチアーキテクチャ (1)

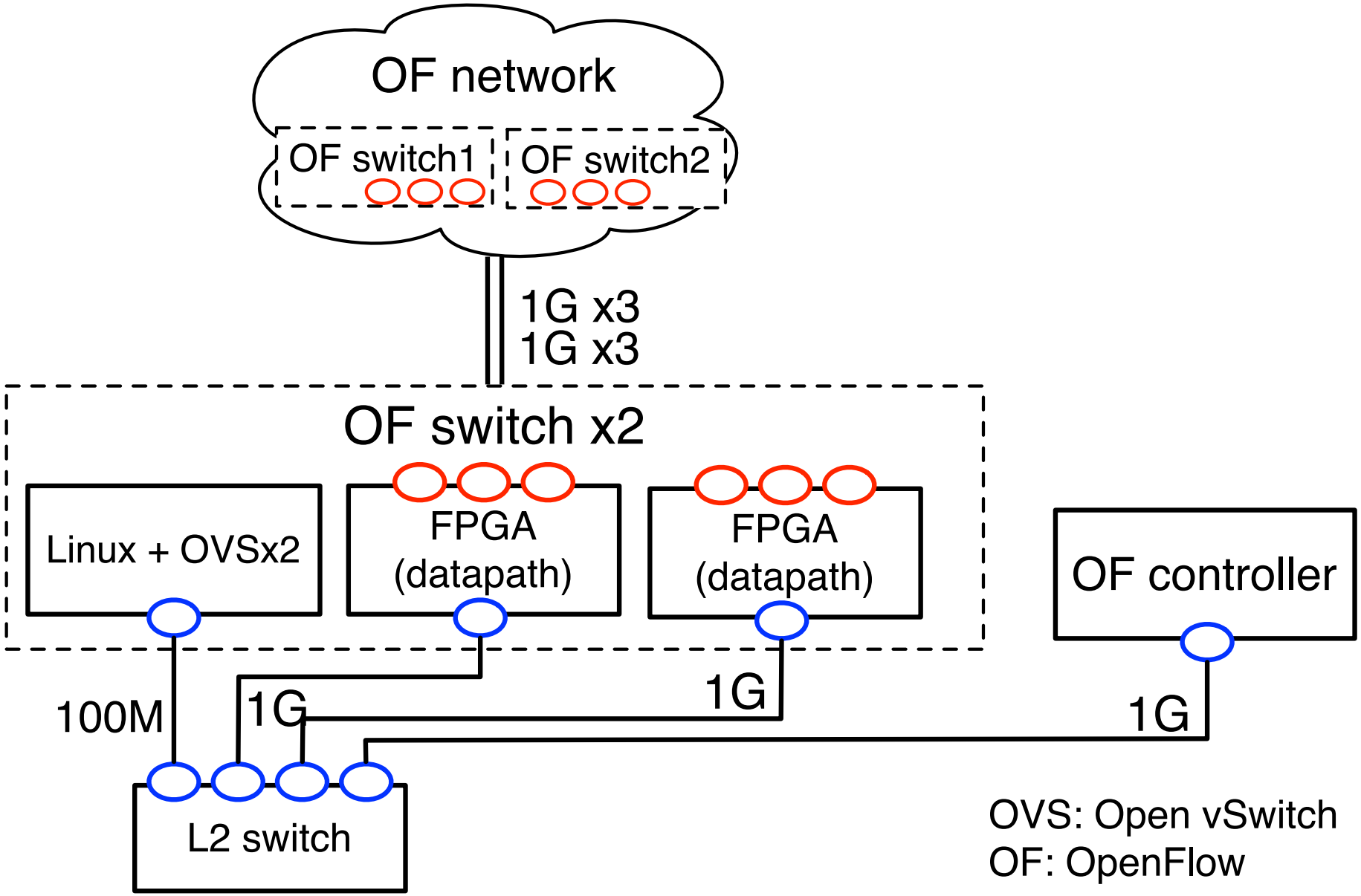
- Linux+Open vSwitch (OVS)のためのHW
- HW化未実装部分はOVSにOffload (NIC機能)
- 自分がほしい機能からHW化



スイッチアーキテクチャ (2)

- Linux+OVS構成だとCPU+MMU, memory, etc.が必要
  - 想定のFPGAのソフトコアだと機能が限定
- OpenFlow SwitchのCPU+SW部と転送用HW部を分離
  - Ethernetを共有バスとして利用




OF network
OF switch1
OF switch2
1G x3
1G x3
OF switch x2
Linux + OVSx2
FPGA
(datapath)
FPGA
(datapath)
OF controller
100M
1G
1G
1G
L2 switch
OVS: Open vSwitch
OF: OpenFlow

スイッチアーキテクチャ (3)

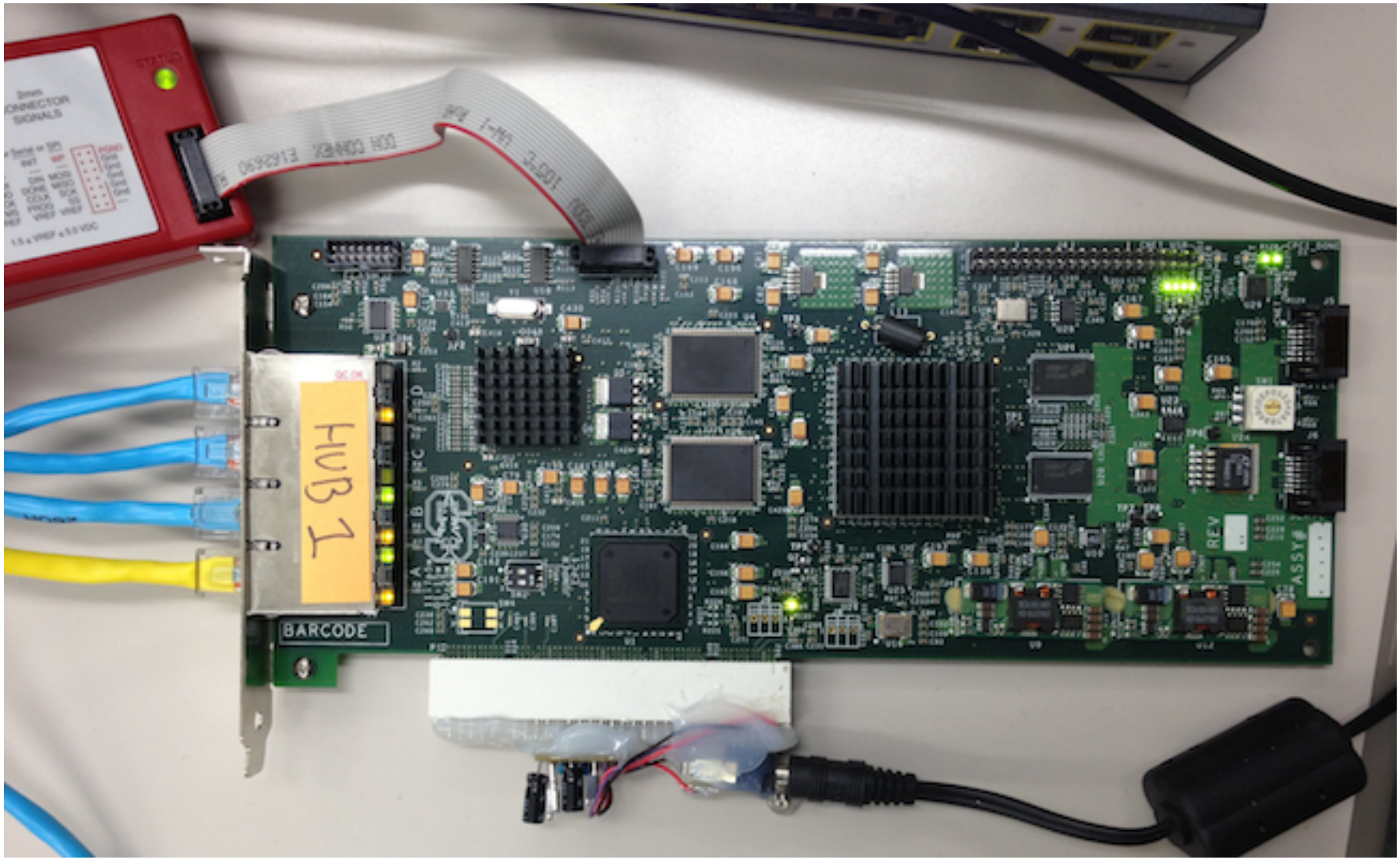
- Cut-through forwarding && Pipeline processing
  - フレームサイズで遅延の変わらない設計
  - OpenFlowの適応領域では遅延が重要なケースが多い
    - ストレージ, VoIP，VMマイグレーション

実装

- Running code!

- 市販MAC IP Coreを使わずに最低限の機能は動作
  - 10/100Base-*，半二重などは未実装
- 開発環境はNetFPGA-1G (1000BASE-T 4ポート)
  - Verilog HDL+オープンソースな開発/検証環境を利用
    - MacでVerilogシミュレーション可能 (iverilog,gtkwave)
  - NetFPGAフレームワークは使わずに新規設計
- code: https://github.com/sora/ovs-hw

転送HW (NetFPGA-1G)

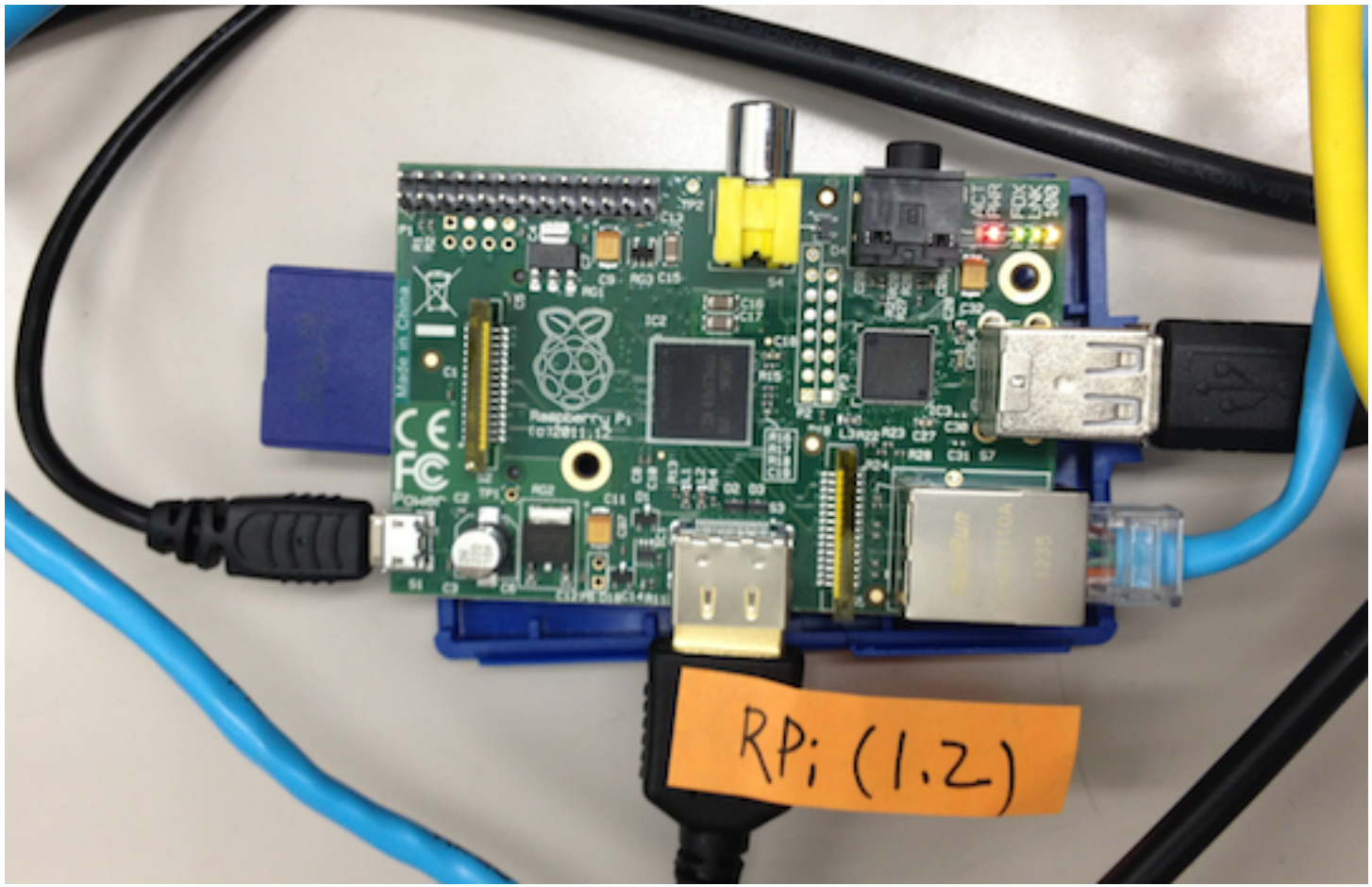


性能測定

- 性能測定 1: 単体の転送性能測定
- 性能測定 2: bonding

- Switch controllerにRaspberry Piを利用
  - $35で買えるARM11 SoCマシン(100 Mbps x1ポート)
  - OpenFlow controllerとのTCP通信などはRaspberryPiにOffload

Linux+Open vSwitch (Raspberry Pi)



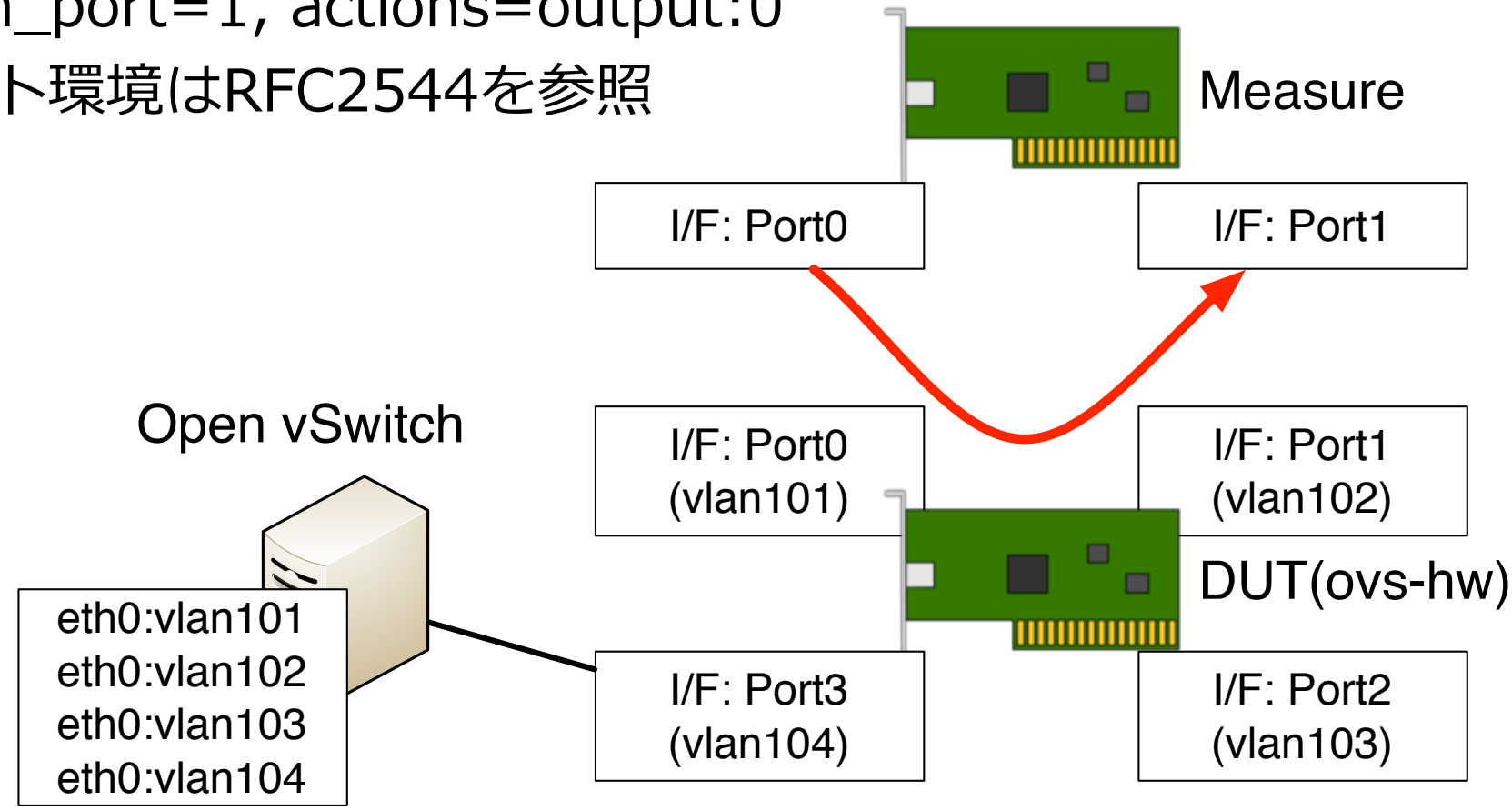
性能測定1: 検証トポロジ

Open vSwitch側ルール

‘in_port=0, actions=output:1’

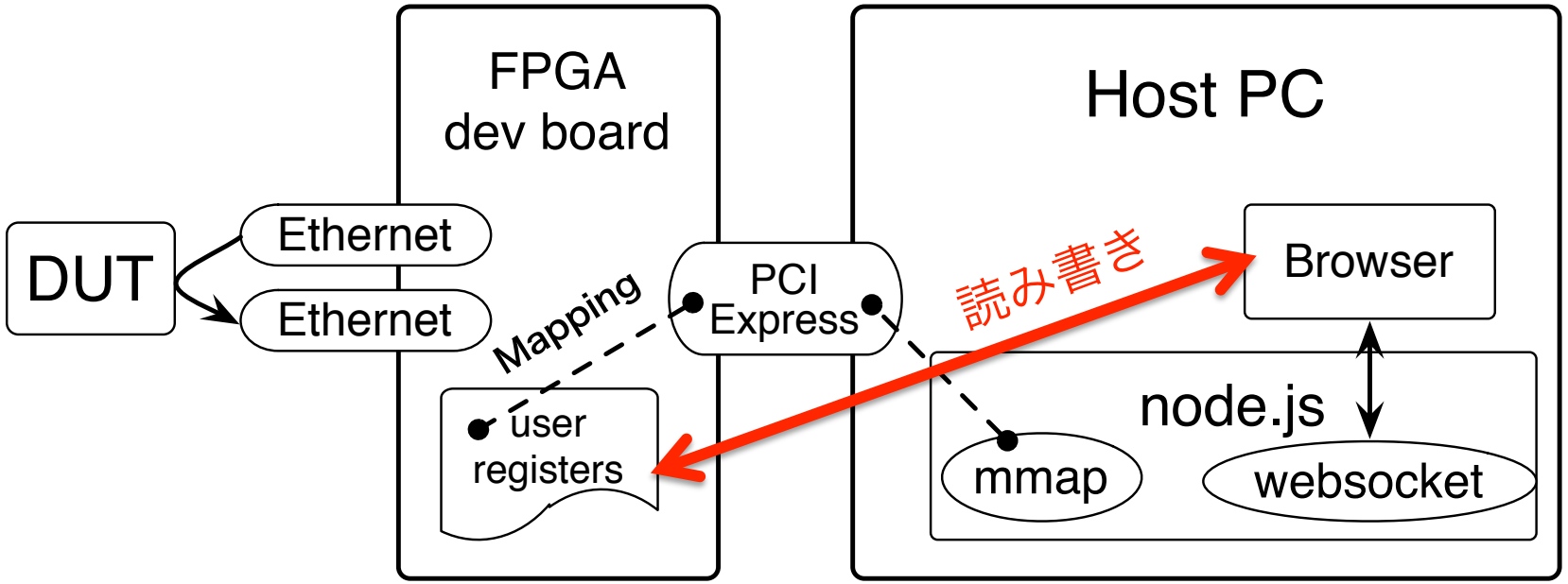
‘in_port=1, actions=output:0’

テスト環境はRFC2544を参照

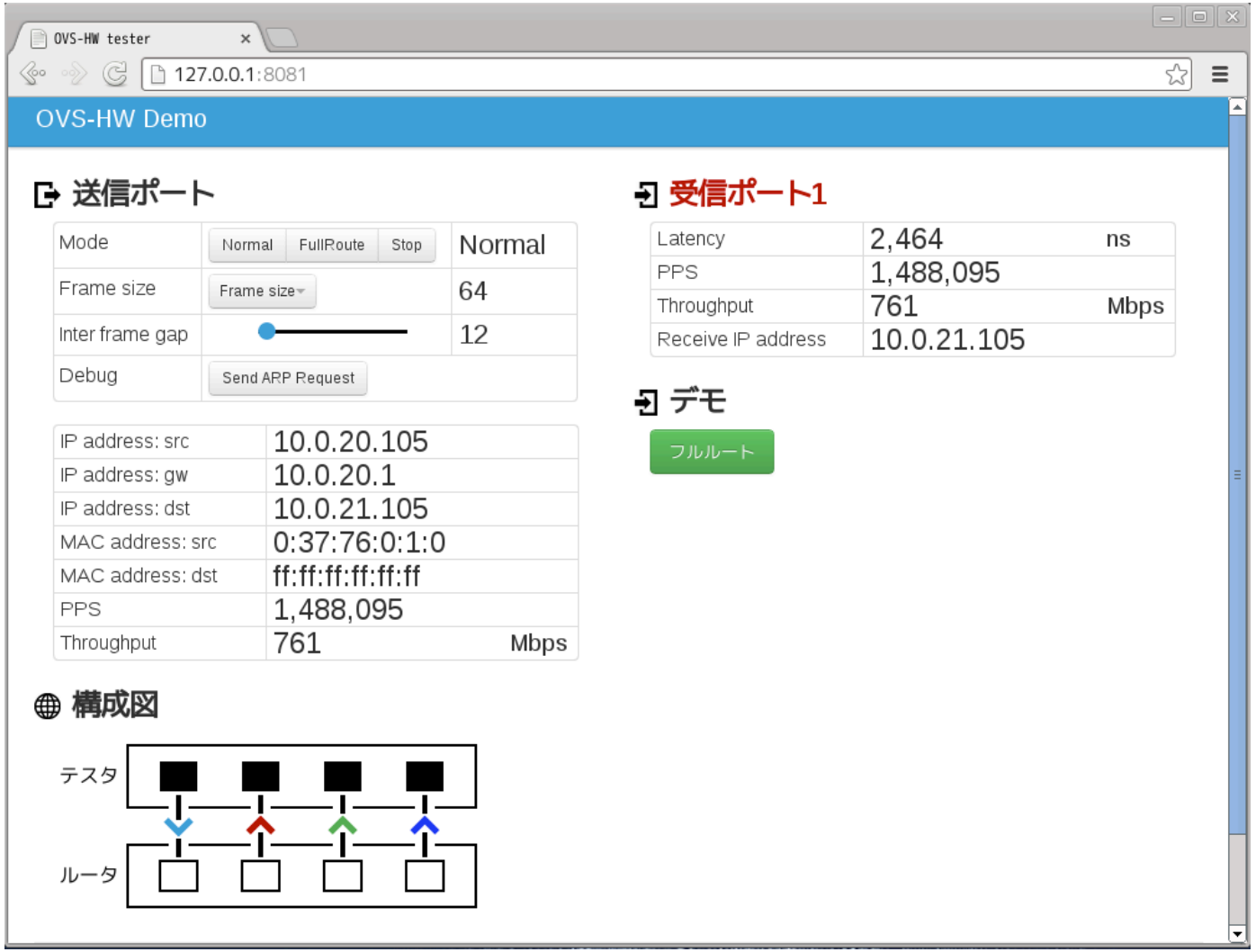


性能測定1: ネットワークテスタの紹介

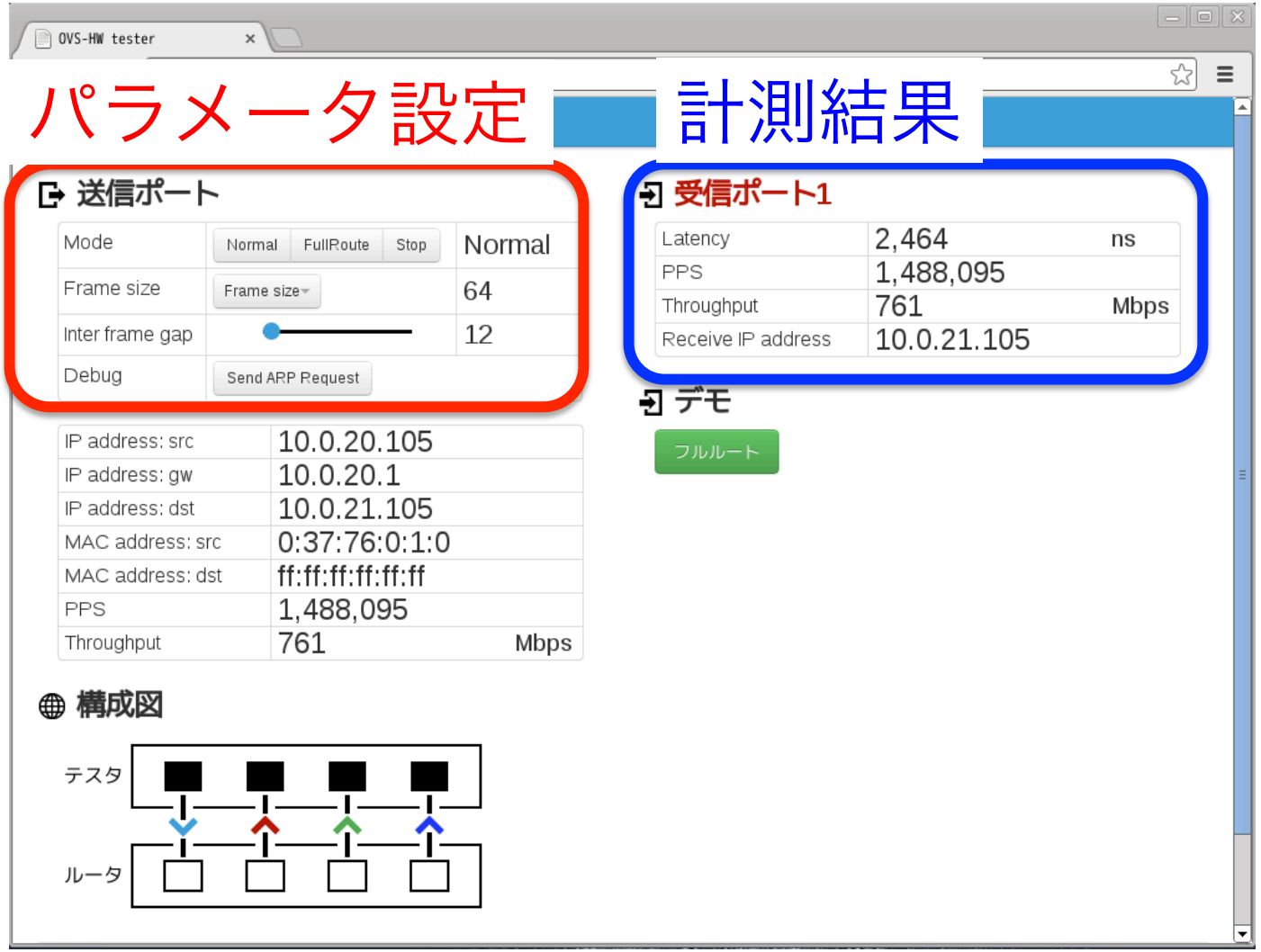
- 手作りFPGA+PCIeネットワークテスタで性能計測
  - Lattice ECP3 versa kitとNetFPGA-1Gで動作
  - Node.jsとwebsocketを利用したwebフロントエンド
- Code:https://github.com/Murailab-arch/magukara/



テスタ画面

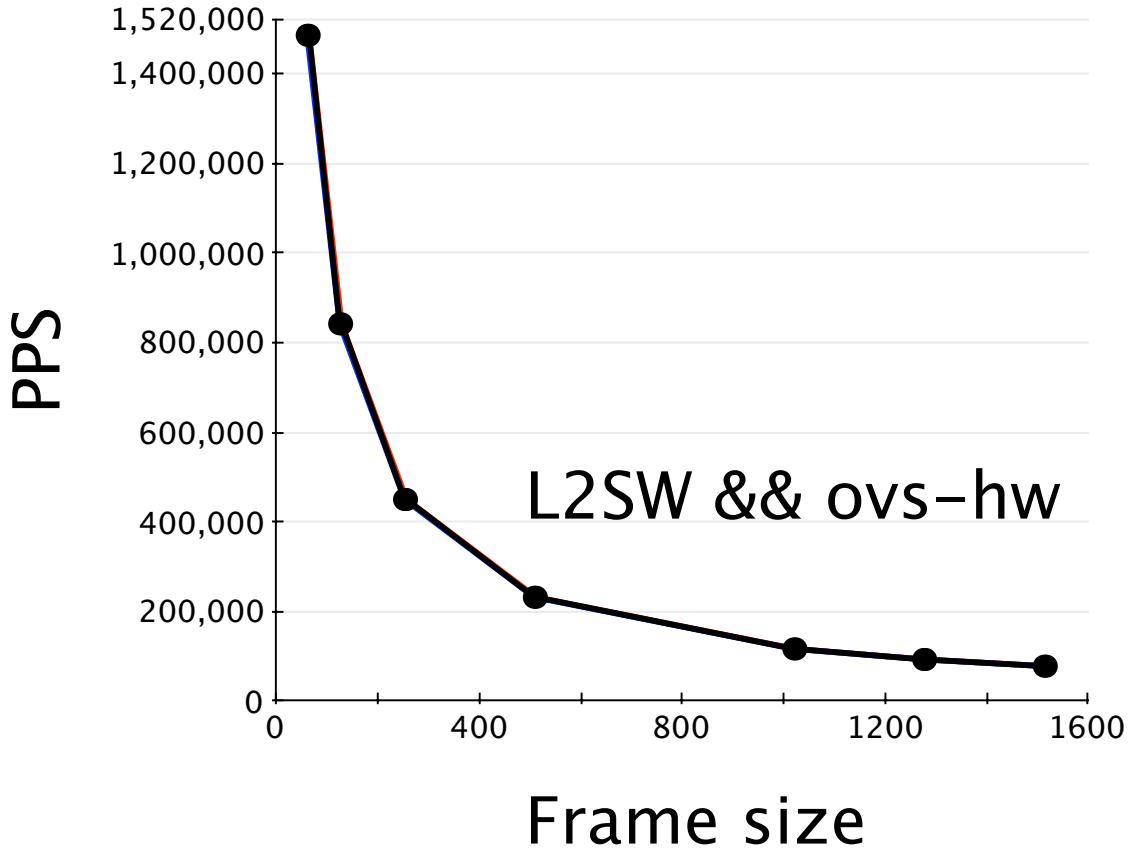

OVS-HW tester
127.0.0.1:8081
OVS-HW Demo
送信ポート
Mode
Normal
FullRoute
Stop
Normal
Frame size
Frame size
64
Inter frame gap
12
Debug
Send ARP Request
IP address: src
10.0.20.105
IP address: gw
10.0.20.1
IP address: dst
10.0.21.105
MAC address: src
0:37:76:0:1:0
MAC address: dst
ff:ff:ff:ff:ff:ff
PPS
1,488,095
Throughput
761
Mbps
構成図
テスタ
ルータ
受信ポート1
Latency
2,464
ns
PPS
1,488,095
Throughput
761
Mbps
Receive IP address
10.0.21.105
デモ
フルルート

テスタ画面


OVS-HW tester
パラメータ設定
計測結果
送信ポート
Mode
Normal
FullRoute
Stop
Normal
Frame size
Frame size
64
Inter frame gap
12
Debug
Send ARP Request
受信ポート1
Latency
2,464
ns
PPS
1,488,095
Throughput
761
Mbps
Receive IP address
10.0.21.105
デモ
フルルート
IP address: src
10.0.20.105
IP address: gw
10.0.20.1
IP address: dst
10.0.21.105
MAC address: src
0:37:76:0:1:0
MAC address: dst
ff:ff:ff:ff:ff:ff
PPS
1,488,095
Throughput
761
Mbps
構成図
テスタ
ルータ

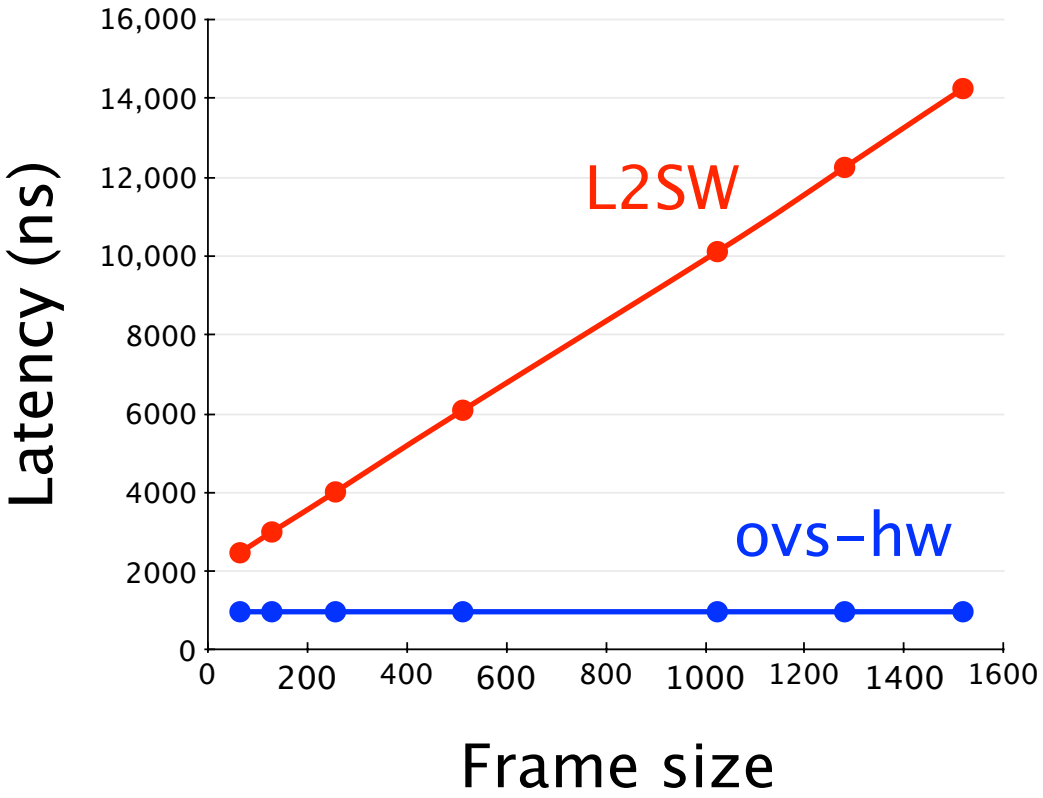
計測結果: PPS計測

- 3000円のL2スイッチとPPS比較



計測結果: 遅延計測

- 3000円のL2スイッチと遅延比較



性能測定2

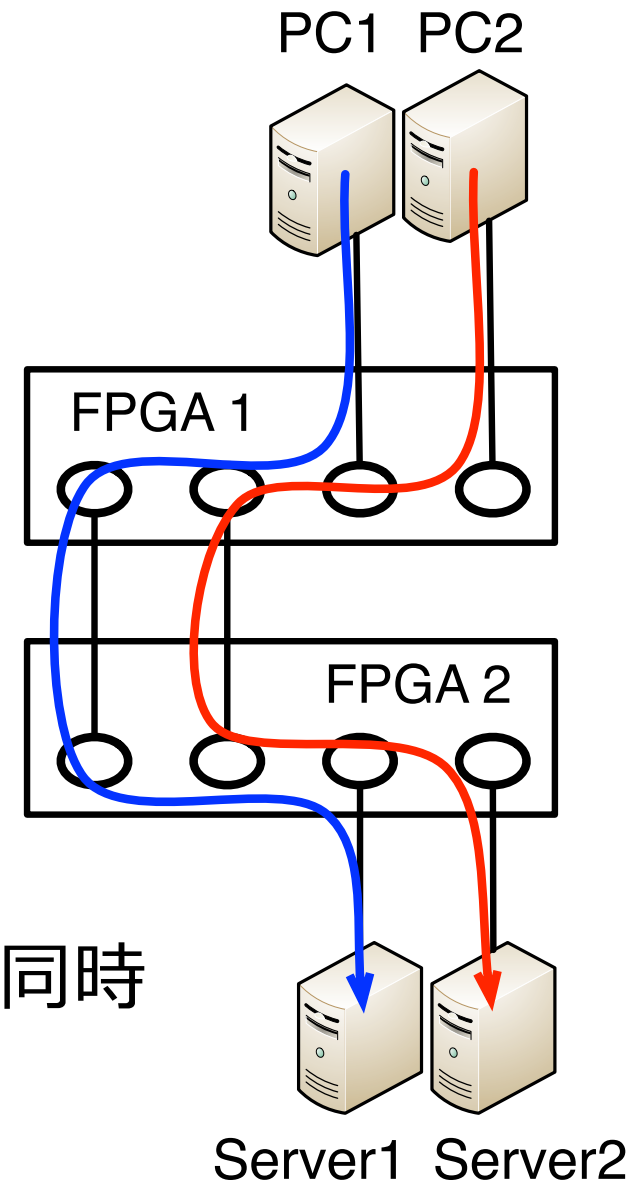
Bondingの性能確認

デバイスドライバが未実装なので，OpenWRTを参考に
　802.1qによる動作検証

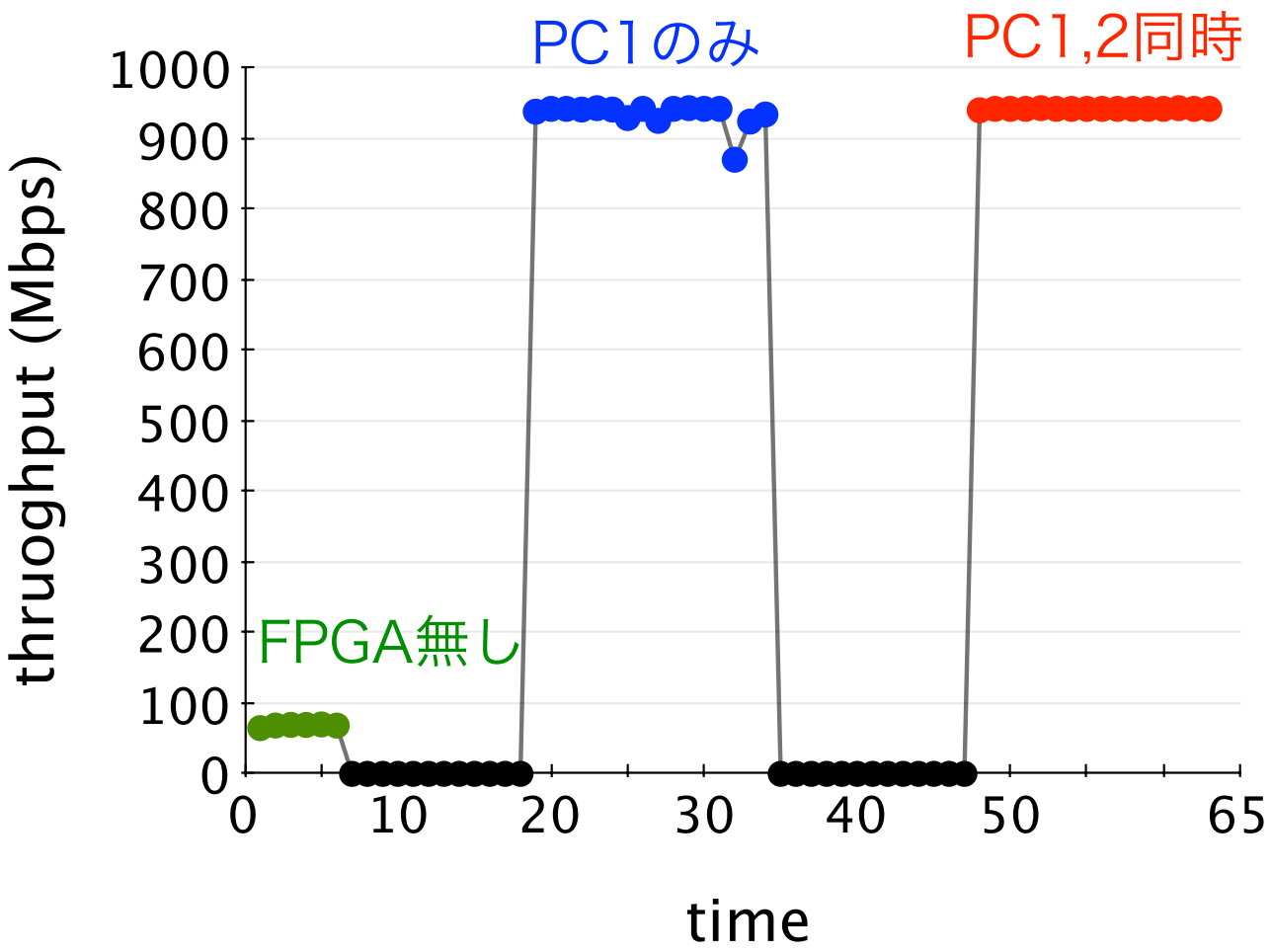
Iperfでスループットを計測

トポロジ図

- フロールール
  - Dest MAC address でoutputを指定
- 計測シナリオ
  - FPGA Offload無し (RPiのみ)
  - PC1-Server1間のみ
  - PC1-SErver1 + PC2-Server2同時



計測結果


PC1のみ
PC1,2同時
FPGA無し
thruoghput (Mbps)
1000
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
65
time

Next step: "FPGA hub"

- 楽しい筐体作り
  - FPGAチップ+メモリ+物理ポートたくさんだけの箱
  - 部品代は計1万5千円から2万円を想定

- 妄想構成 (価格は発注数量1の場合)
  - FPGA (XC6SLX45T) 5,000 yen
  - RJ45 x8
  - PHY chip x8
  - SRAM (QDRII) 4 MB 1,500 yen
  - Hub間接続 (SATA 3Gbps x4)

まとめ

- 高機能なソフトウェアスイッチであるOpen vSwitchのためのHWアーキテクチャを提案
- Open vSwitchをHW化+物理ポート拡張する箱を検討

- 今後について
  - Flow tableやstatistics保持方法の検討
  - デバイスドライバ, Openflowプロトコルの連携
  - OVSとの連携強化 (ofproto_classとの連携など)
  - OpenFlow以外のOVS機能を検討

Q and A

code:

ovs-hw, https://sora.github.com/ovs-hw/

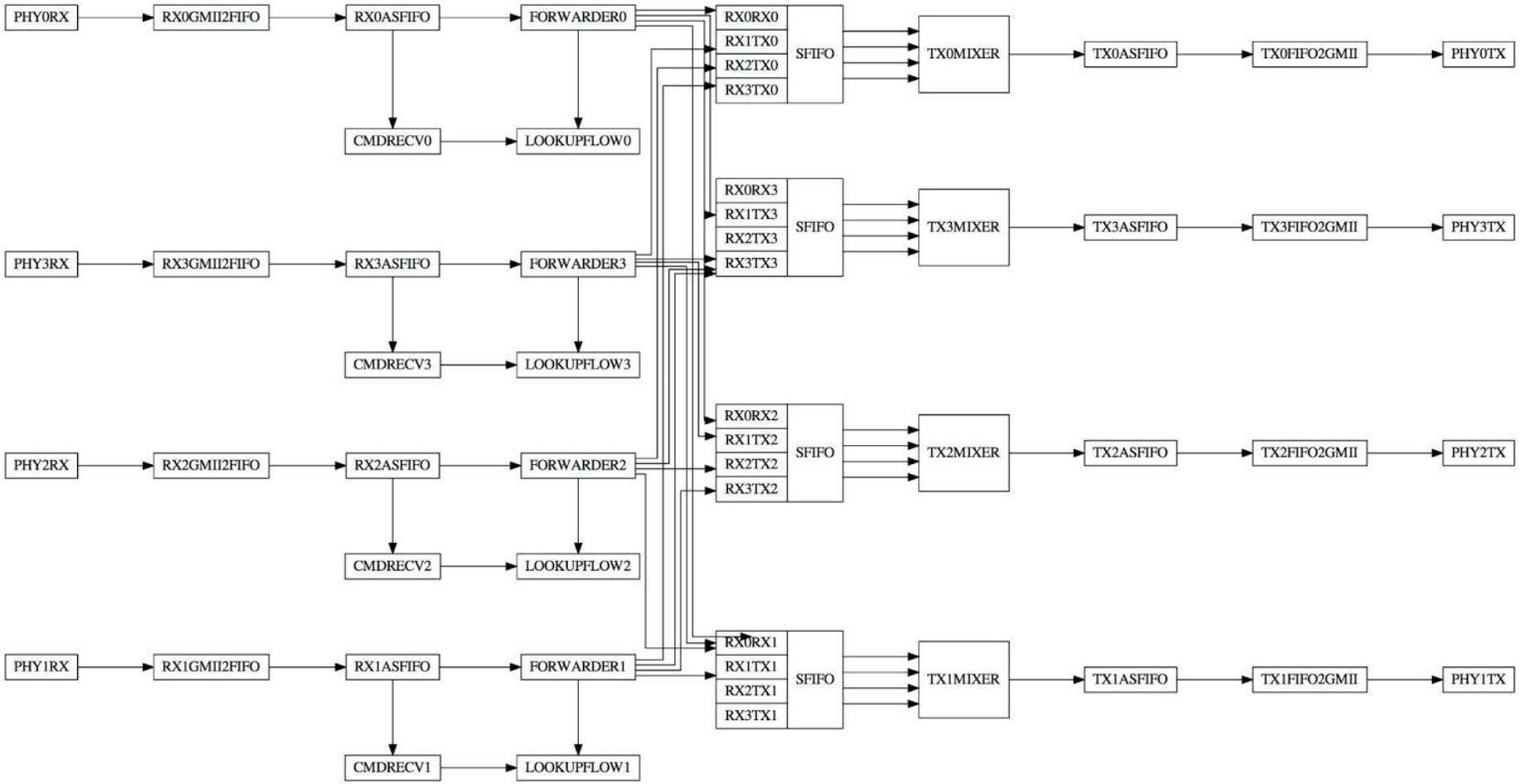
magukara,https://github.com/Murailab-arch/magukara/

Simulation on your Mac:

1. brew install icarus-verilog gtkwave
2. git clone http://github.com/sora/ovs-hw
3. cd ovs-hw; make test

補足スライド

モジュール構成図



https://github.com/sora/ovs-hw/tree/master/doc/block_diagram

マルチポート10Gデバイス DIYの検討

- 実際は開発，検証環境を揃えるのが困難
- 参考までに可能性だけ検討

- 開発キットが続々登場
  - Xilinx Kintex7 connectivity kit, NetFPGA-10Gなど
- 回路: 1000BASE-Tとの違い (10G-*Rの場合)
  - PHYチップの機能をFPGAで処理
    - SERDESを使った最低限のPCS/PMA機能
    - 回路規模が大きいため上位モデルのFPGAが必要
    - もし個人で買える10 GbE PHYがあると便利
  - XGMII対応(125MHz, 8 bit -> 156.25MHz, 64 bit)

Flow table, statistics回路 案

- Flow table
  - いくつかの組み合わせを検討 (使用用途で入れ替え)
    - TCAM (BlockRAM) + Hash (SRAM) など
  - その他
    - FPGAなので未使用のtupleはコードをコメントアウトして，その分のリソースをFlow tableに回す

- Statistics
  - HP DevoFlowなどのトライアルを参考に検討
    - サンプリング，タイミング調整，など

スイッチアーキテクチャ

- 仮想ポートと物理ポートのマッピング方法
- 独自Ethernet type +
  802.1Qライクなカプセル化
- 現在はデバドラが未実装なので
  ポートごとに別々のVLAN IDを
  付与して識別