ITU Kaleidoscope 2016

ICTs for a Sustainable World

Multi-path Chunked Video Exchanges over OF@TEIN SDN Cloud Playground

Phyo May Thet*, JongWon Kim✧, Chaodit Aswakul*

*Wireless Network and Future Internet (WIFUN) Research Group
Department of Electrical Engineering, Faculty of Engineering
Chulalongkorn University, Bangkok, Thailand

✧Network Computing Systems Laboratory (NetCS)
School of Information and Communications,
Gwangju Institute of Science and Technology, South Korea

Bangkok, Thailand
14-16 November 2016

We are …

Ms. Phyo May Thet
phyo.ma@student.chula.ac.th

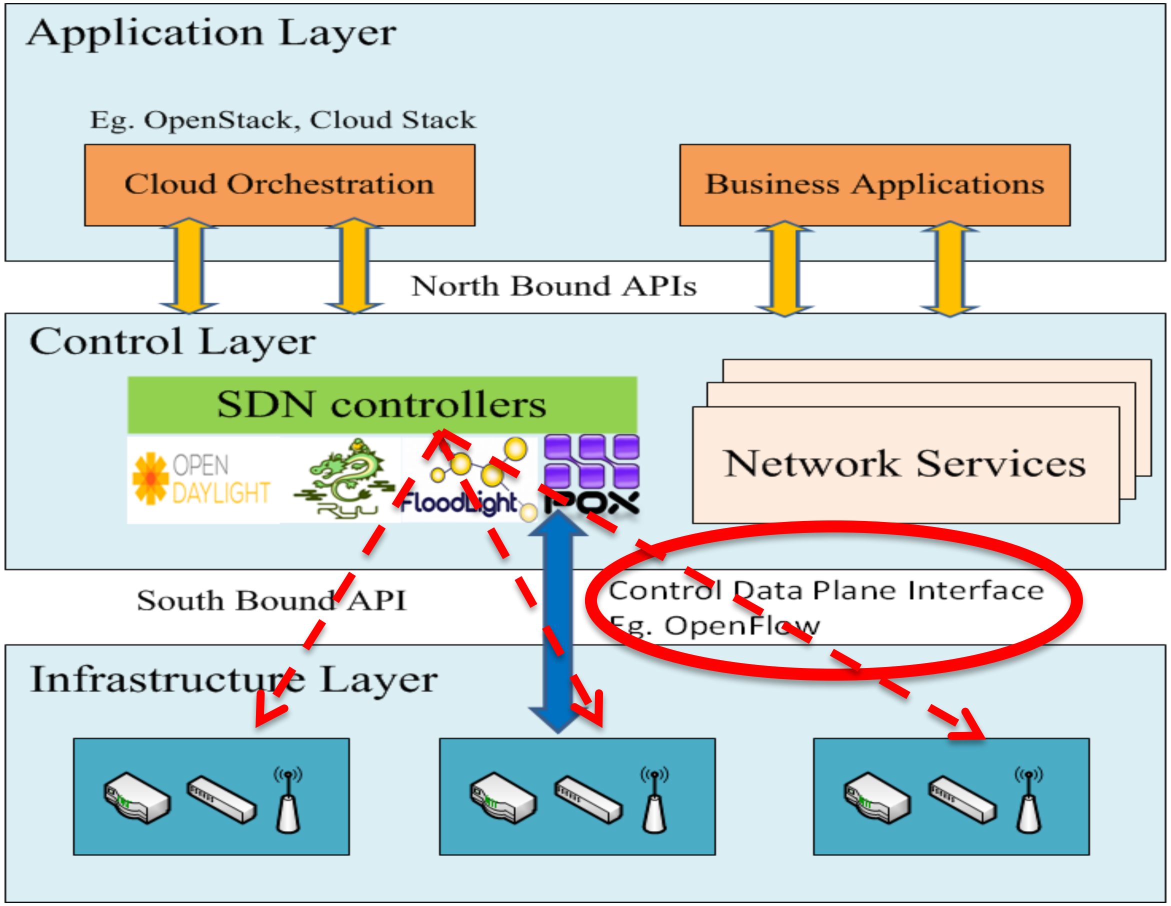
Prof. JongWon Kim
jongwon@nm.gist.ac.kr

Assoc. Prof. Dr. Chaodit Aswakul
chaodit.a@chula.ac.th

Outlines of Talk

- Why do we need Software-Defined Networking (SDN)?
- Introduction of OF@TEIN SDN Cloud Playground Testbed
- Motivation and objectives
- Design of Multi-path Chunked Video Exchanges (i.e., file transferring and streaming)
- Results and Discussion
- Summary

Software-Defined Networking (SDN)



Application Layer

Eg. OpenStack, Cloud Stack

Cloud Orchestration

Business Applications

North Bound APIs

Control Layer

SDN controllers

Network Services

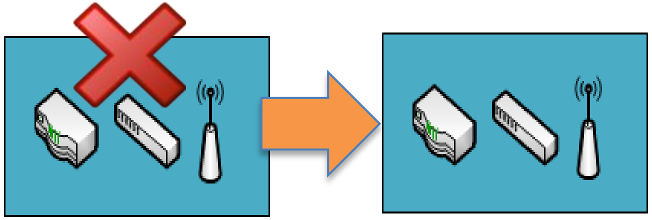
South Bound API

Control Data Plane Interface
Eg. OpenFlow

Infrastructure Layer

- Enable direct communication between SDN controller and networking devices

Closed System **Open System**

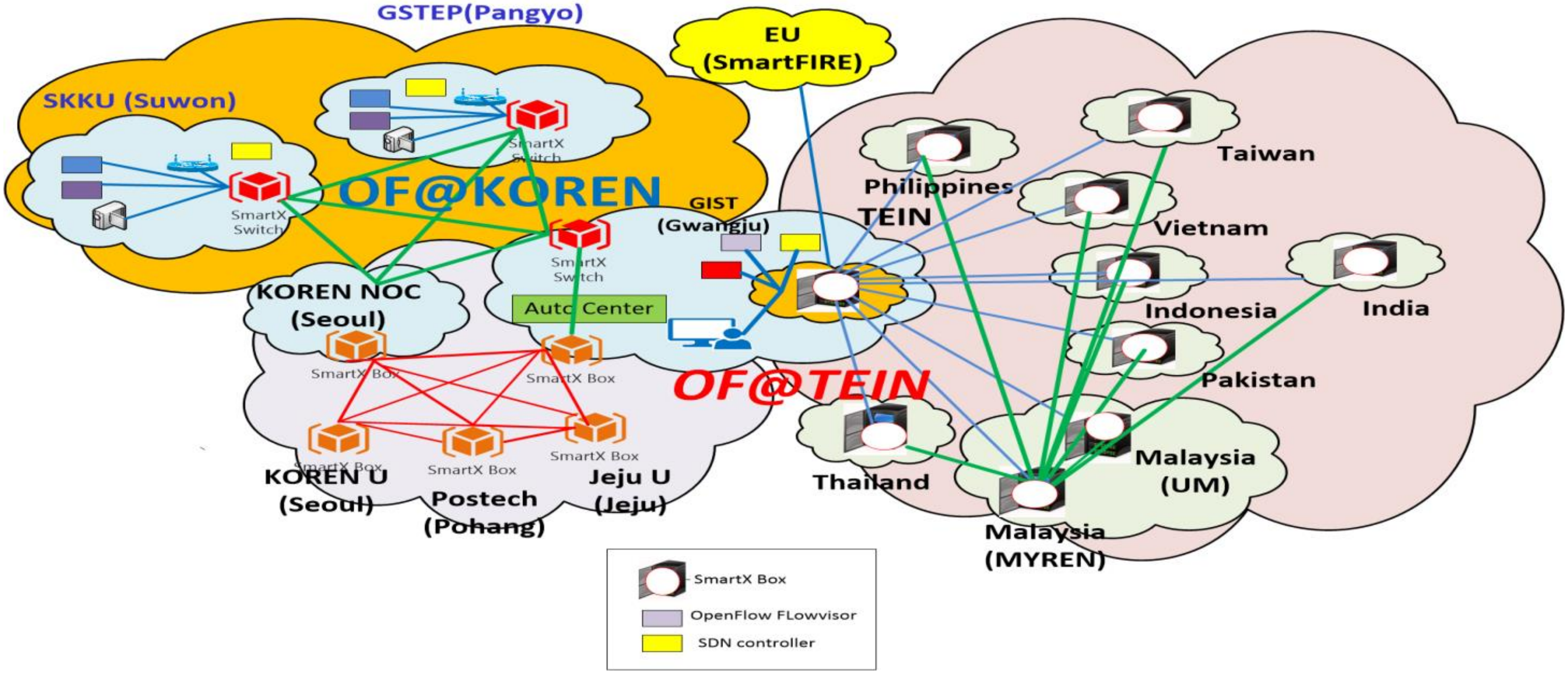


[1] ONF Market Education Committee, and others, "Software defined networking: The new norm for networks," *ONF White Paper.*, Palo Alto, US, Open Networking Foundation, April, 2012.
[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review 38.*, pp. 69-74, April, 2008.

OF@TEIN Infrastructure*

connected to 11 sites in 9 countries

Updated (2015-07-31)



*Use OF@TEIN infrastructure fig. permitted by Aris

[3] A.C. Risdianto, N.L. Kim, J. Shin, J. Bae, M. Usman, T.C. Ling, P. Panwaree, P.M. Thet, C. Aswakul, N.H. Thanh, A. Iqbal, U. Javed, M.U. Ilyas, and J. Kim, “OF@TEIN: A community efforts towards open/shared SDN-Cloud virtual playground,” *in Proc. of the Asia-Pacific Advanced Network 40*., pp. 22-28, August 10-14, 2015.

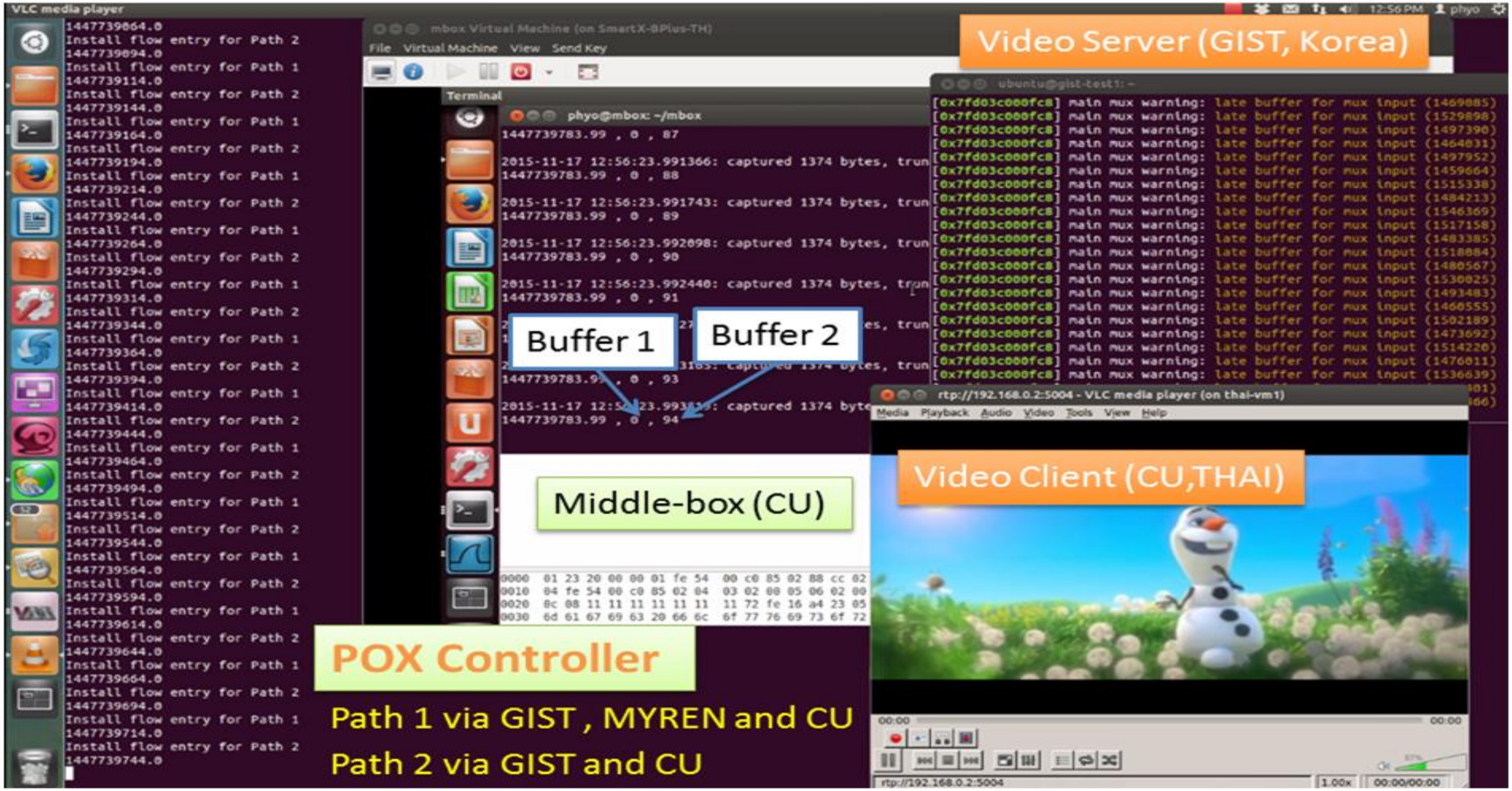
Objectives

- To design and develop middle-box splitting functionalities for chunked video exchanges (i.e., file transfer and streaming) by using POX controller, OpenStack, Open vSwitches, KVM based Middle-box and SmartX boxes
- To reduce middle-box processing delay and to enable multi-path capacity leverage by introducing the combination of multi-path file transfer function and Tsunami protocol[4] over OF@TEIN SDN cloud playground
- To investigate cloud file transferring and video streaming over OF@TEIN SDN cloud playground

[4]M.R. Meiss, "Tsunami: A high-speed rate-controlled protocol for file transfer," *Indiana University*, 2004.

Why Multi-path Chunked Video Exchanges?

- Motivation to reduce middle-box processing delay (demo)


VLC media player
mbox Virtual Machine (on SmartX-BPlus-TH)
File Virtual Machine View Send Key
Terminal
phyo@mbox: ~/mbox
ubuntu@gist-test1: ~
rtp://192.168.0.2:5004 - VLC media player (on thai-vm1)
Media Playback Audio Video Tools View Help
rtp://192.168.0.2:5004
1.00x
00:00/00:00
12:56 PM
phyo

Video Server (GIST, Korea)

Buffer 1

Buffer 2

Middle-box (CU)

Video Client (CU,THAI)

POX Controller

Path 1 via GIST , MYREN and CU

Path 2 via GIST and CU

```
1447739864.0
Install flow entry for Path 2
1447739894.0
Install flow entry for Path 1
1447739114.0
Install flow entry for Path 2
1447739144.0
Install flow entry for Path 1
1447739164.0
Install flow entry for Path 2
1447739194.0
Install flow entry for Path 1
1447739214.0
Install flow entry for Path 2
1447739244.0
Install flow entry for Path 1
1447739264.0
Install flow entry for Path 2
1447739294.0
Install flow entry for Path 1
1447739314.0
Install flow entry for Path 2
1447739344.0
Install flow entry for Path 1
1447739364.0
Install flow entry for Path 2
1447739394.0
Install flow entry for Path 1
1447739414.0
Install flow entry for Path 2
1447739444.0
Install flow entry for Path 1
1447739464.0
Install flow entry for Path 2
1447739494.0
Install flow entry for Path 1
1447739514.0
Install flow entry for Path 2
1447739544.0
Install flow entry for Path 1
1447739564.0
Install flow entry for Path 2
1447739594.0
Install flow entry for Path 1
1447739614.0
Install flow entry for Path 2
1447739644.0
Install flow entry for Path 1
1447739664.0
Install flow entry for Path 2
1447739694.0
Install flow entry for Path 1
1447739714.0
Install flow entry for Path 2
1447739744.0
```

```
1447739783.99 , 0 , 87

2015-11-17 12:56:23.991366: captured 1374 bytes, trun
1447739783.99 , 0 , 88

2015-11-17 12:56:23.991743: captured 1374 bytes, trun
1447739783.99 , 0 , 89

2015-11-17 12:56:23.992098: captured 1374 bytes, trun
1447739783.99 , 0 , 90

2015-11-17 12:56:23.992440: captured 1374 bytes, trun
1447739783.99 , 0 , 91

2                          27                    es, trun
1

2                          3165: captured 1374 bytes, trun
1447739783.9  , 0 , 93

2015-11-17 12:5  23.993   : captured 1374 byte
1447739783.99 , 0 , 94
```

```
0000  01 23 20 00 00 01 fe 54  00 c0 85 02 88 cc 02
0010  04 fe 54 00 c0 85 02 04  03 02 00 05 06 02 00
0020  0c 08 11 11 11 11 11 11  11 72 fe 16 a4 23 05
0030  6d 61 67 69 63 20 66 6c  6f 77 76 69 73 6f 72
```

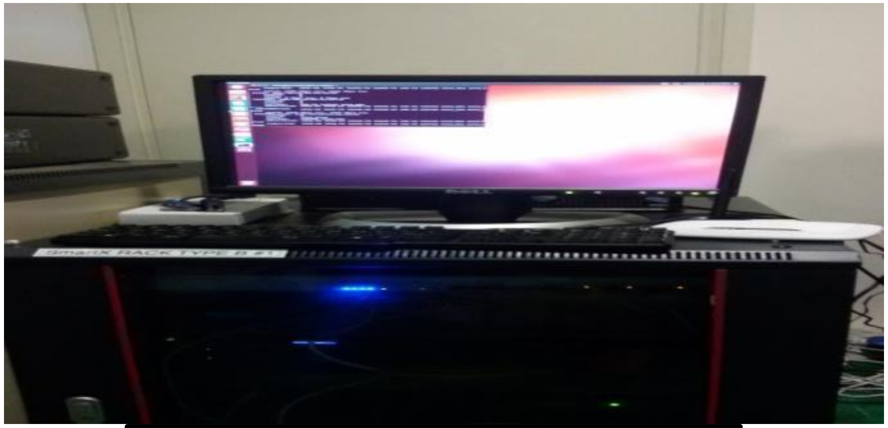
```
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1469885)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1529898)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1497390)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1464031)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1497952)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1459664)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1515338)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1484213)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1546369)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1517158)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1483385)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1518884)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1480567)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1530025)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1493483)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1460555)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1502189)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1473692)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1514220)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1476011)
[0x7fd03c000fc8] main mux warning: late buffer for mux input (1536639)
                                                              01)
                                                              66)
```

OF@TEIN SmartX boxes location

Server room @Chamchuri 9 Building

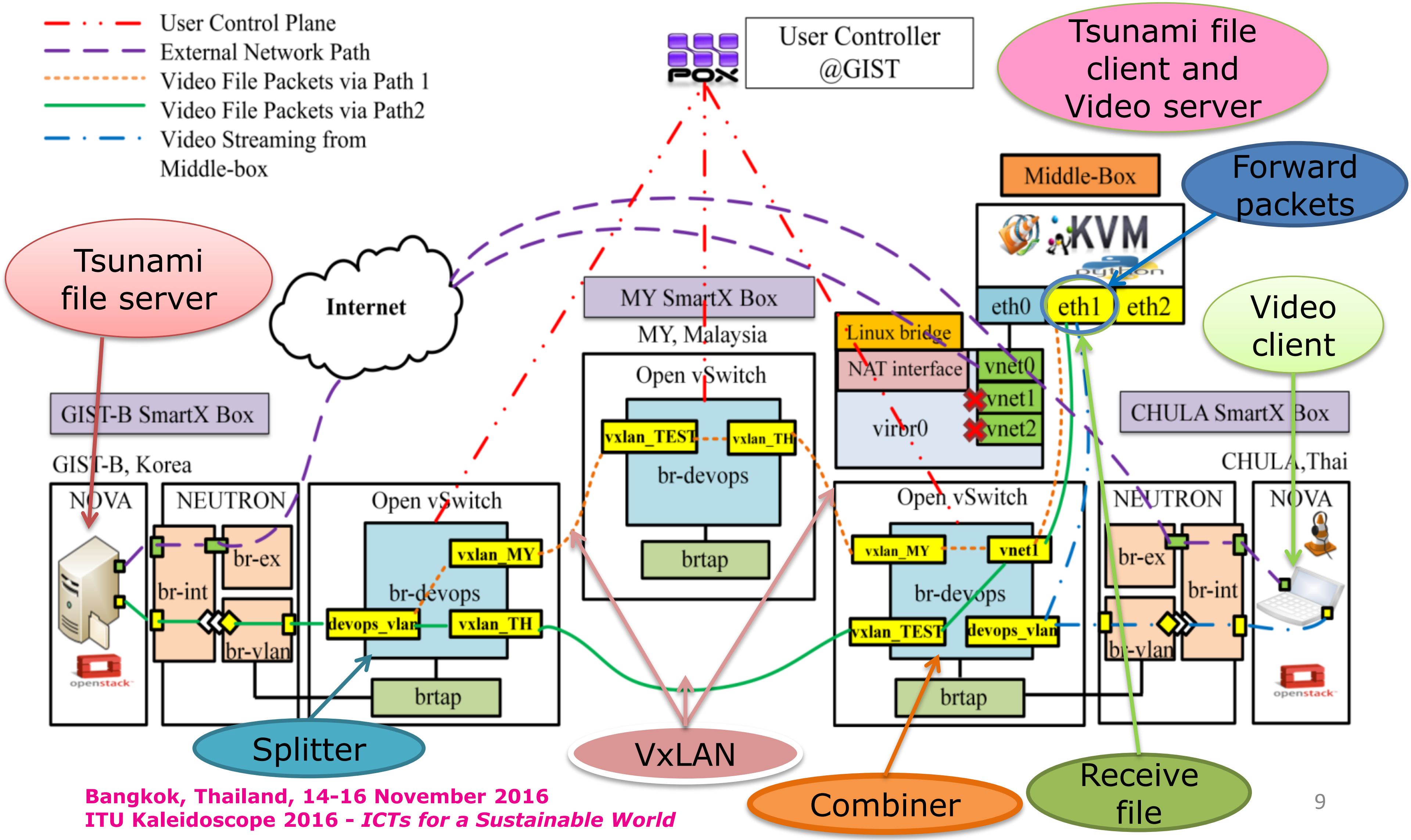
CHULA SmartX Box

Server room @GIST



GIST-B SmartX Box

Architecture of Multi-path Chunked Video Exchanges over OF@TEIN SDN Cloud Playground



Tsunami file client and Video server

Forward packets

Tsunami file server

Video client

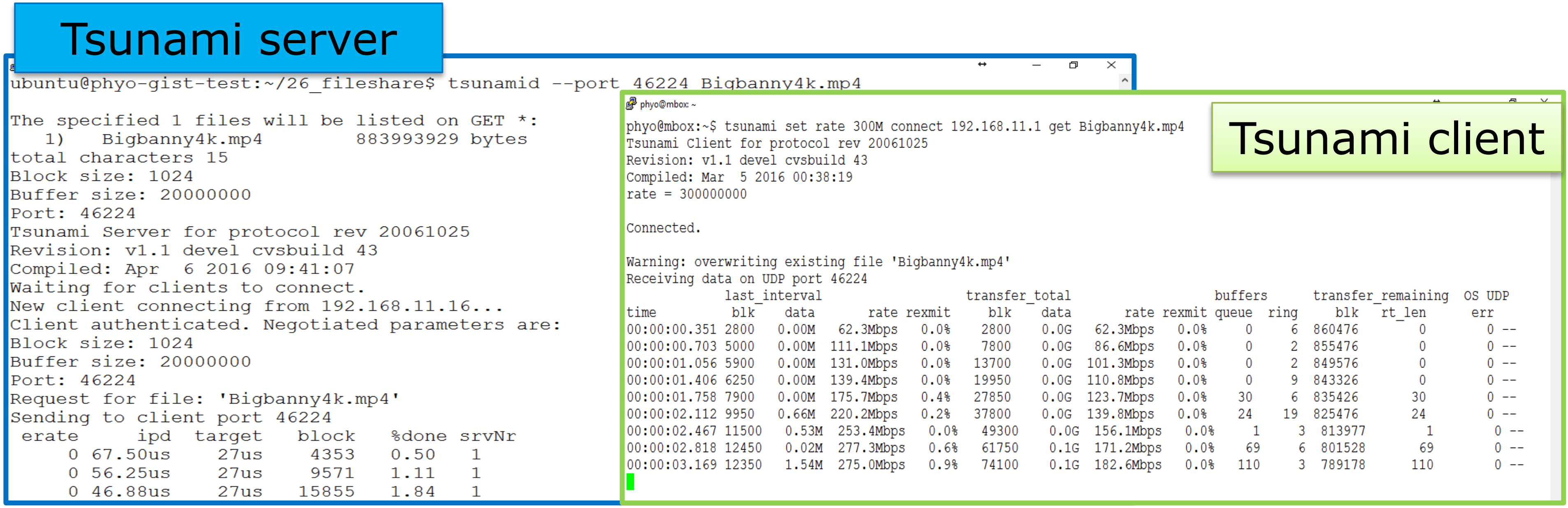
Splitter

VxLAN

Combiner

Receive file

Video File Transfer Experimental Settings



Tsunami server

```
ubuntu@phyo-gist-test:~/26_fileshare$ tsunamid --port 46224 Bigbanny4k.mp4

The specified 1 files will be listed on GET *:
   1)    Bigbanny4k.mp4         883993929 bytes
total characters 15
Block size: 1024
Buffer size: 20000000
Port: 46224
Tsunami Server for protocol rev 20061025
Revision: v1.1 devel cvsbuild 43
Compiled: Apr  6 2016 09:41:07
Waiting for clients to connect.
New client connecting from 192.168.11.16...
Client authenticated. Negotiated parameters are:
Block size: 1024
Buffer size: 20000000
Port: 46224
Request for file: 'Bigbanny4k.mp4'
Sending to client port 46224
 erate       ipd    target     block     %done  srvNr
     0 67.50us      27us      4353    0.50     1
     0 56.25us      27us      9571    1.11     1
     0 46.88us      27us     15855    1.84     1
```

Tsunami client

```
phyo@mbox:~$ tsunami set rate 300M connect 192.168.11.1 get Bigbanny4k.mp4
Tsunami Client for protocol rev 20061025
Revision: v1.1 devel cvsbuild 43
Compiled: Mar  5 2016 00:38:19
rate = 300000000

Connected.

Warning: overwriting existing file 'Bigbanny4k.mp4'
Receiving data on UDP port 46224
             last_interval                   transfer_total                   buffers      transfer_remaining  OS UDP
time          blk    data        rate rexmit     blk     data        rate rexmit queue   ring     blk   rt_len      err
00:00:00.351 2800   0.00M   62.3Mbps   0.0%    2800    0.0G   62.3Mbps   0.0%     0      6  860476       0        0 --
00:00:00.703 5000   0.00M  111.1Mbps   0.0%    7800    0.0G   86.6Mbps   0.0%     0      2  855476       0        0 --
00:00:01.056 5900   0.00M  131.0Mbps   0.0%   13700    0.0G  101.3Mbps   0.0%     0      2  849576       0        0 --
00:00:01.406 6250   0.00M  139.4Mbps   0.0%   19950    0.0G  110.8Mbps   0.0%     0      9  843326       0        0 --
00:00:01.758 7900   0.00M  175.7Mbps   0.4%   27850    0.0G  123.7Mbps   0.0%    30      6  835426      30        0 --
00:00:02.112 9950   0.66M  220.2Mbps   0.2%   37800    0.0G  139.8Mbps   0.0%    24     19  825476      24        0 --
00:00:02.467 11500   0.53M  253.4Mbps   0.0%   49300    0.0G  156.1Mbps   0.0%     1      3  813977       1        0 --
00:00:02.818 12450   0.02M  277.3Mbps   0.6%   61750    0.1G  171.2Mbps   0.0%    69      6  801528      69        0 --
00:00:03.169 12350   1.54M  275.0Mbps   0.9%   74100    0.1G  182.6Mbps   0.0%   110      3  789178     110        0 --
```

- Resolution : 3840 x 2610(4-K video); Duration : 10 min
- Total file size : 843 Mbytes (6744 Mbits)

Tsunami file transfer protocol (TCP+UDP)

- Bulk data are transferred via UDP and control data are transferred via TCP

Tsunami file transfer parameters @GIST-B Tsunami server

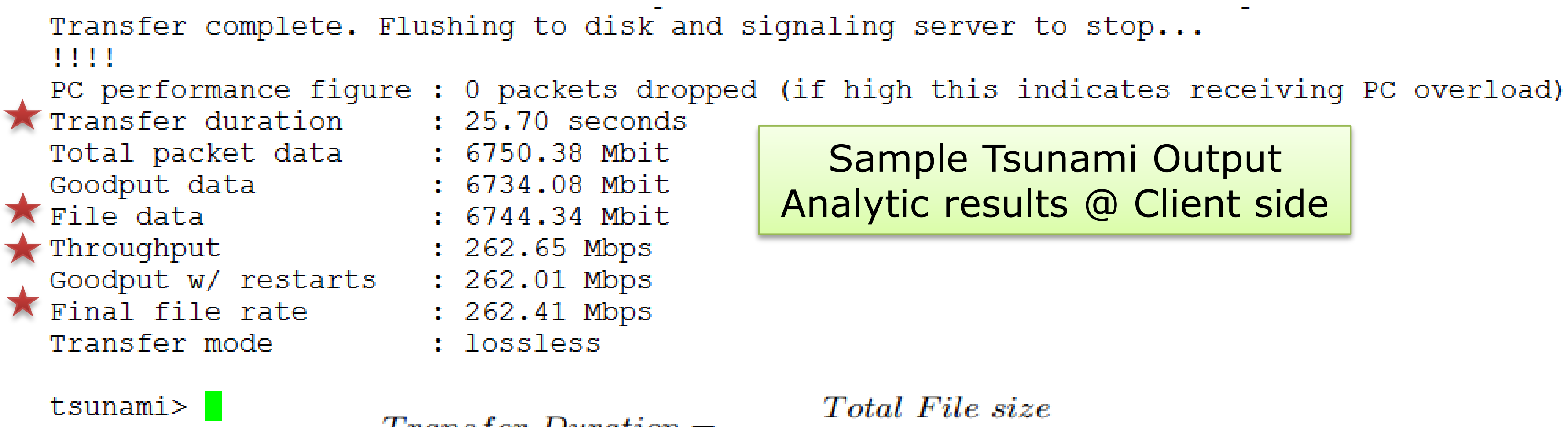
- Block size:1024 bytes (how large UDP blocks to use)
- Buffer size: 20 Mbytes (size of ring buffer in RAM)

Tsunami file transfer parameters @CHULA Middle-box Tsunami client

- Target file transfer rate: 100,200,300,400,500 Mbits/s

Experimental Scenarios

Scenario	Selected parameter for Tsunami
Multi-path (1 sec via MY: 2 sec via GIST-B)	Target file transfer rate: 100, 200, 300, 400, 500 Mbps
Single-path (GIST-B>MY> CHULA)	
Single-path (GIST-B> CHULA)	



```
  Transfer complete. Flushing to disk and signaling server to stop...
  !!!!
  PC performance figure : 0 packets dropped (if high this indicates receiving PC overload)
★ Transfer duration     : 25.70 seconds
  Total packet data     : 6750.38 Mbit
  Goodput data          : 6734.08 Mbit
★ File data             : 6744.34 Mbit
★ Throughput            : 262.65 Mbps
  Goodput w/ restarts   : 262.01 Mbps
★ Final file rate       : 262.41 Mbps
  Transfer mode         : lossless

  tsunami> █
```

Sample Tsunami Output
Analytic results @ Client side

$$Transfer\ Duration = \frac{Total\ File\ size}{Actual\ File\ Transfer\ Rate}$$

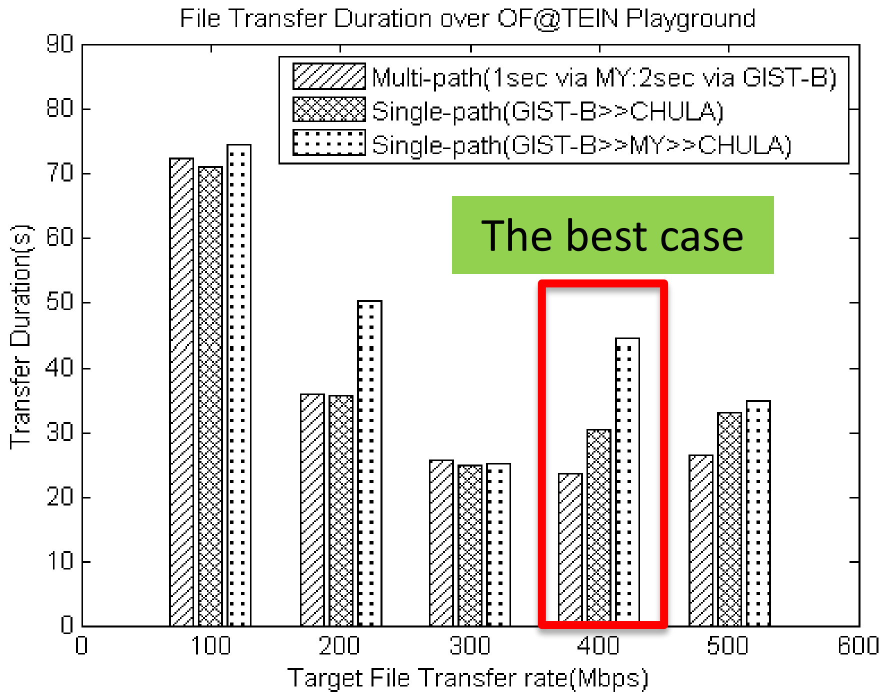
where: $Transfer\ Duration$ = File transfer duration between server and client in seconds (s),

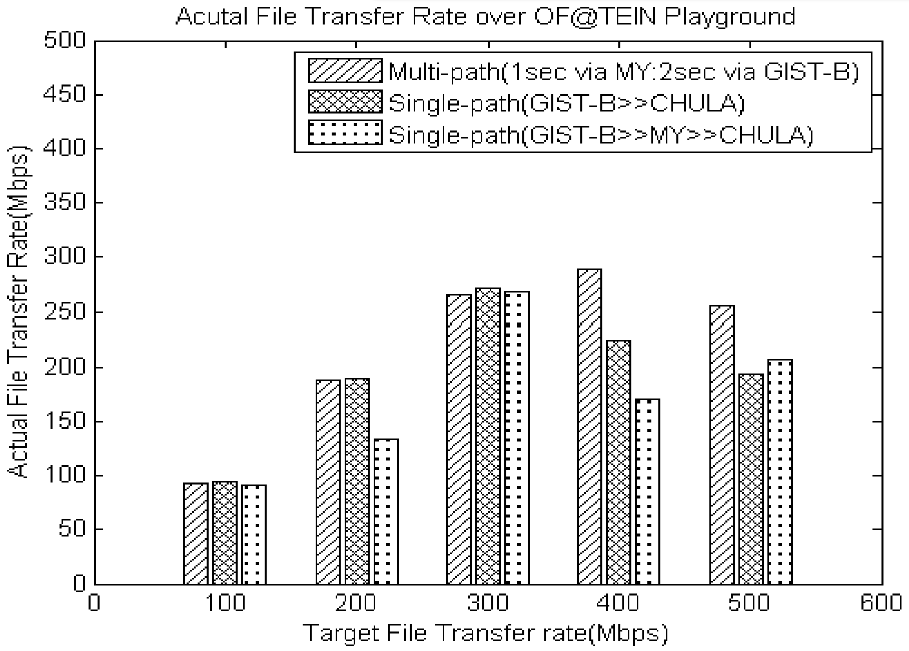
$Total\ File\ size$ = File size in bits (bits) and

$Actual\ File\ Transfer\ Rate$ = Rate of file transmission in bits per second (bps).

Results and Discussion

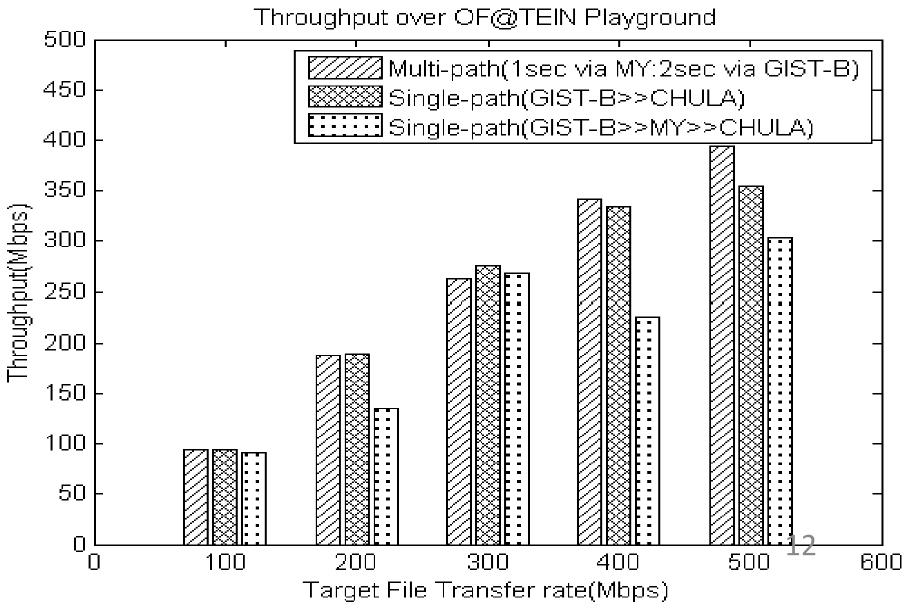
Actual file transfer rate





File transfer duration

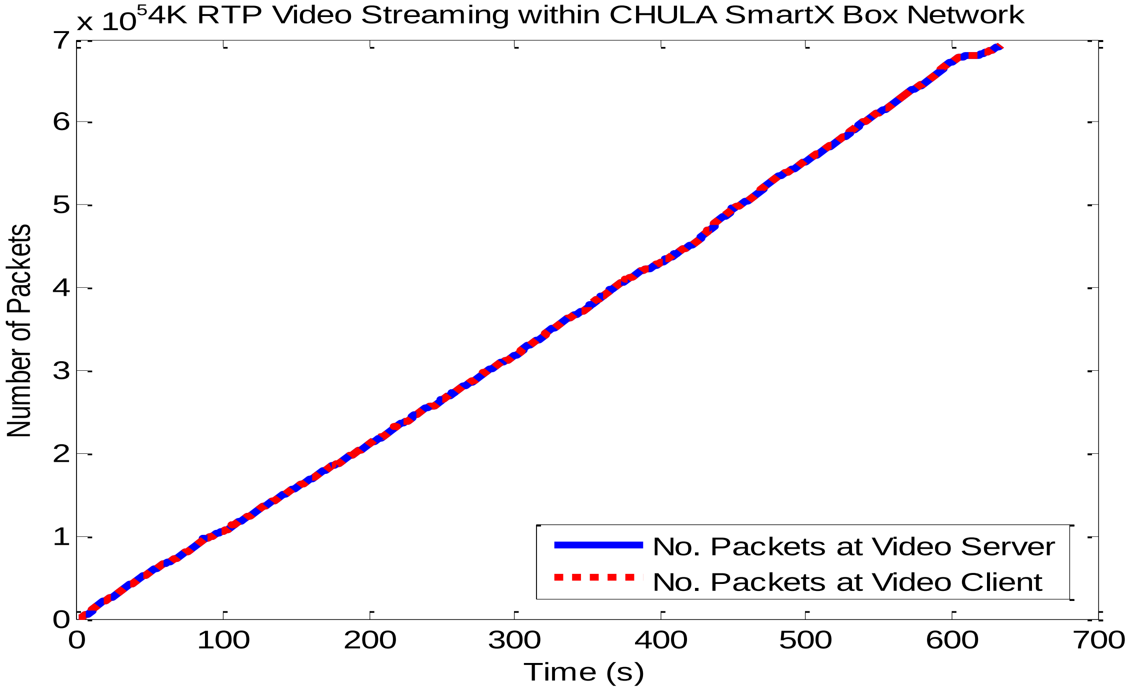
Throughput



Path1 , UDP throughput : 422 Mbps
& RTT:~125ms
Path2 , UDP throughput : 456 Mbps
& RTT:~105ms

4K RTP Video Streaming within Chula SmartX Box Network

- To investigate the performance of 4K RTP video streaming within a local area network



- Packet loss ratio- 0%
- Less packet delay compares to the streaming over international links

- Resolution   : 3840x2610; Duration : 10 min
- Video Codec : H.264 ; Audio and Video Mean bit rate – 11133 kbits/s
- Streaming Mode: RTP
- VLC player is used for both server and client over X 11 Desktop environment

Summary of Multi-path Chunked Video Exchanges over OF@TEIN SDN Cloud Playground

- ➢ We have tested the combination of traditional Tsunami protocol and proposed multi-path file transferring function. And streamed out the 4K video streaming within CHULA SmartX Box network.
- ➢ We have implemented the X11 desktop environment and access method for remote OpenStack VMs in order to use the GUI applications with fast access.

Recommended	Not recommended
When the links are congested and not enough to carry the whole video file traffic.	When the main path capacity already suffices for carrying out the incoming packets of video file traffic and the main path already having a lower RTT delay than other available paths
When the selected target file transfer rate of Tsunami protocol lead to be congested on the available network links.	
To adjust the video resolution scale to be around 50% (eg. resolution:1630x937) lower than the normal 4K resolution scale due to VLC player bottleneck for 4K video	

Thank you

Assoc. Prof. Dr. Chaodit Aswakul
Wireless Network & Future Internet Research Group
Department of Electrical Engineering
Chulalongkorn University, THAILAND
chaodit.a@chula.ac.th

[**www.ee.eng.chula.ac.th**](http://www.ee.eng.chula.ac.th)