Evaluation of MTCP over POX Controller

FAID Souad
2G/3G/LTE Network Design & Optimization Engineer, huawei,Casablanca,Morocco

Mohamed Moughit
National School of Applied Sciences,Univ Hassan1, IR2M Laboratory, Khouribga, Morocco

Noureddine IDBOUFKER
National School of Applied Sciences, Laboratory, Marrakech, Morocco

Abstract— Recently, networked systems have become more complex, having more server nodes and being scattered around the world across the wide-area network.
Additionally, many networked systems are multi-homed, that is, is connected to multiple gateway networks leading into the internet. Prevalent examples including mobile phones that can connect to the internet via the cellular network using 3G or 4G and the local-area network using WIFI.

In order to transfer these huge data sets we need to make efficient use of all available network capacity. This means using multiple paths when available. In this paper a prototype of such network is presented. Several emerging network technologies are integrated to achieve the goal of efficient high end-to-end throughput. Multipath TCP is used by the end hosts to distribute the traffic across multiple paths and OpenFlow is used within the network through its controllers which was the subject of my previous paper entitled **"Openflow controllers performance Evaluation"**, to do the wide area traffic engineering. This design should be able to maximize bandwidth and network path utilization by allowing hosts to take advantage of presently-unused paths. And it also should be scalable to benefit distributed file storage systems, data-intensive services, or any high-performance computing systems.

Index Terms—SDN,OPENFLOW,MTCP,TCP, Multiple Path,POX Controller, Traffic Engineering.

1 Introduction

OpenFlow (OF) is considered one of the first software-defined networking (SDN) standards. It originally defined the communication protocol in SDN environments that enables the SDN Controller to directly interact with the forwarding plane of network devices such as switches and routers, both physical and virtual, so it can better adapt to changing business requirements. Multipath TCP is a new approach towards efficient load balancing. Instead of letting the network do the load balancing by using hashes and ECMP, MPTCP is doing the load balancingin the end nodes as part of the TCP process. MPTCP is an extension of the TCP/IP stack.The byte stream of the application is split across multiple subflows, one for each interface. MPTCP can handle paths of different bandwidth because there is a congestion control mechanism across the subflows. Which can move the traffic on a congested path to a link with less congestion. So it adapts the load balancing according to the load of other traffic on the paths.In this paper, MPTCP will be used in combination with an OpenFlow based multipath network. Figure 1 shows an example of such a network. The topology has multiple paths of various capacity between the servers. The four switches are controlled via theOpenFlow protocol by an OpenFlow application.This application automatically detects the topology by listening to LLDP (link Layer Discovery Protocol) packets (received via theOpenFlow protocol) and calculates link disjoint paths between the servers. The forwarding entries that are needed for the paths are then pushed to the switches. [1]

- *FAID Souad is with Laboratory of Computer Networks, Mobility and Modeling IR2M National School of Applied Sciences,Morocco (faidsouad@gmail.com) MOUGHIT*
- *Mohamed Laboratory of Computer Networks,Mobility and Modeling IR2M National School of Applied Sciences University Hassan First Settat, Morocco(m.moughit@gmail.com).*

2 Multiphating concepts

Multipathing in communication networks is gaining momentum due to its attractive features of increased reliability, throughput, fault tolerance, and load balancing capabilities. In particular, wireless environments and datacenters are envisioned to become largely dependent on the power of multipathing for seamless handovers, virtual machine (VM) migration and in general, pooling less proficient resources together for achieving overall high proficiency. The transport layer, with its knowledge about end-to-end path characteristics, is well placed to enhance performance through better utilization of multiple paths.

Three major benefits obtained by using multiple paths are mentioned below.

1) Load Balancing: Load balancing is traditionally performed at the network layer, wherein a network operator can reroute traffic in order to avoid congested hotspots. However, traffic engineering at the network layer can be unstable and may lead to oscillations [2].

2) Resource Pooling: The idea of pooling resources together into one single resource, which has the aggregate abilities of all the resources, has been widely used in the internet [3].

3) Diversification: Diversity is a technique that is frequently deployed in datacenters, wireless environments, and the Internet to improve performance. By exploiting the diversity in the characteristics of multiple paths, performance enhancements including bandwidth aggregation and reliability can be achieved.

2.1MPTCPprotocol

Multipath Transmission Control Protocol, or MPTCP is an extension to TCP at transport layer to utilize multiple paths between two network endpoints by stripping data into multiple subflows.Each subflow then behaves like a TCP flow, with its own congesion control, send and receive windows, and so on. Multipathing allows us to use more than one path in one logical connection, increasing bandwidth utilization, improving redundancy and stability, as well as allowing seamless handovers in certain environments. By decomposing the transport layer into two sublayers as in figure 2, MPTCP can separately recognize the end-to-end and point-to-point situations better than the traditional model by having the upper half, which is the MPTCP extension, work only with managing the connection and subflows, while the lower half works like ordinary TCP, dealing with congestion and other matters in each subflow.[4]

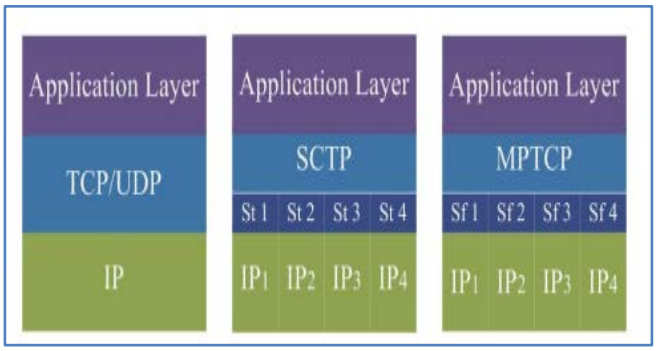


Fig.1. Architecture of UDP, TCP, SCTP and MPTCP

MPTCP is designed to work with extant internet architecture. However, it was not easy to deploy a new protocol in the Internet as it consists of various devices including proxies and firewalls that are designed to stop any suspicious flow.

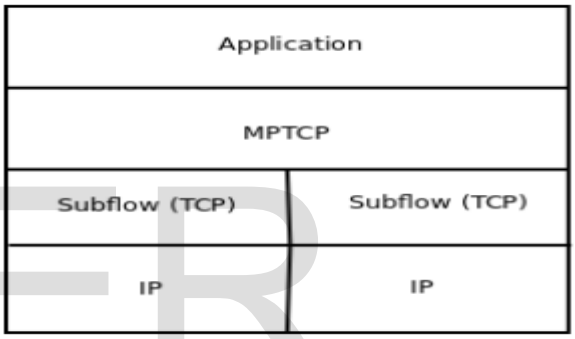


Fig.2. MPTCParchitecture

Some of the problems that might be able to hinder the operation of any new protocol such as MPTCP are:

- Middleboxes: They work with current TCP implementation but they can terminate any connection or drop packets if they think packets are not in right order or it is not a right packet.
- Compatibility with non-MPTCP host: How MPTCP at one end host will behave if other end does not recognize MPTCP protocol. As it is tough to upgrade all the software at all devices at once.
- TCP friendly: Since, it will coexist with single path TCP, it should not take away the bandwidth that single path TCP is supposed to use.
- At least as secure and robust as single TCP. Since, it is an upgradation; everyone wants to have a better security or at least as good as previous version if not better.
- Throughput: Improve throughput if possible otherwise provide throughput

at least as good as best single path TCP. Otherwise, what's the use, why someone wants to use MPTCP.

- MPTCP should work with both "make-before-break" and "break-before-make".
- Compatible with Applications: MPTCP should expose API to the application, so that they can use it. Additionally it should follow the same service model as provided by TCP i.e. reliable, in-order delivery to the application.[5]

2.2 MTCP functionning

The following figure shows how MPTCP works. In a MPTCP enabled kernel the TCP component is split in a MPTCP component and TCP subflow components for each interface. The MPTCP component receives a byte stream from the application. The MPTCP component implements several functions. It takes care of path management by detecting and using multiple paths to a destination. Packet scheduling splits the byte stream received from the application in multiple segments and transmits these segments on one of the available subflows.
These segments are numbered, so that the receiving MPTCP component can put the segments in the correct order and reconstruct the original byte stream. Finally there is congestion control across the subflows. This function spreads the load over the subflows. When a subflow becomes congested, traffic is moved to a subflow that is less congested. This function also takes care of retransmissions on another subflow when one of the subflows fails.[6]

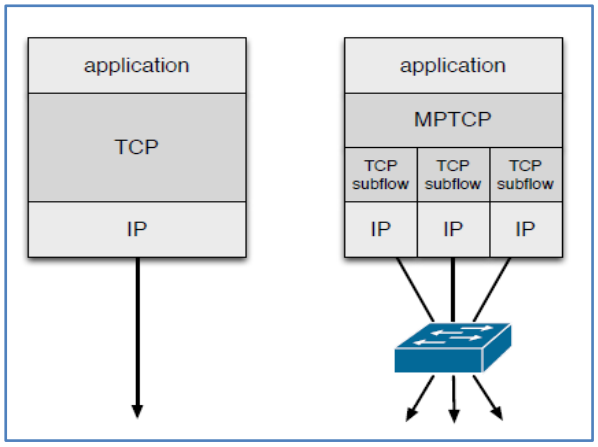
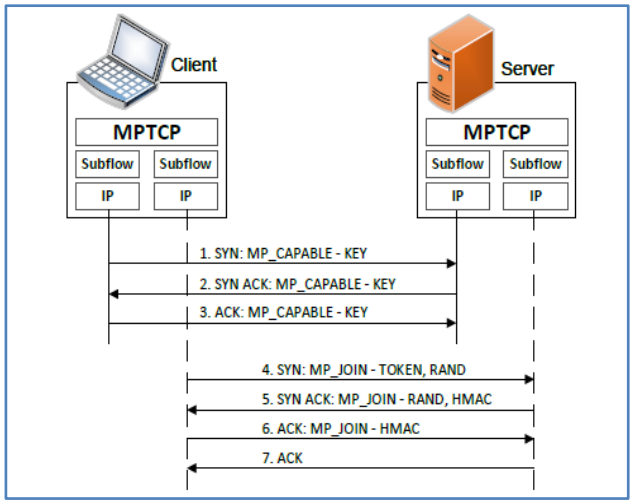


Fig. 3. Traditional TCP vs Multipath TCP

Figure 4 shows the initial connection setup and a subflow connection setup with server A as the initiator of both setups. MPTCP client starts with an initial connection setup that is similar to a normal TCP connection setup with a SYN, SYN/ACK, ACK sequence. The only difference is that an MPTCP capable server adds the MP CAPABLE TCP option. Server A sends a TCP SYN packet with the MP CAPABLE TCP option. This option field also contains its authentication key and some additional flags to indicate whether checksums are required and the cryptographic algorithm to use.

Fig.4. Multipath TCP Connection Setup



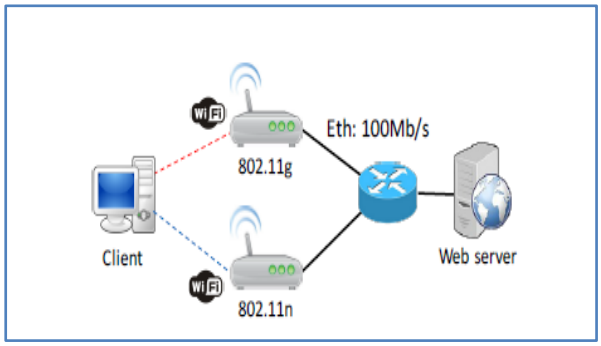
A. Congestion Control in MPTCP

One of the most important aspect of TCP is to recover from congestion. Since many connection might be using the same routers and the paths despite having different end hosts. It's hard to detect congestion and recover from it.
1. MPTCP should give throughput at least as much as single path TCP will give on one of the best path.
2. Multiple subflows of the same flow should not take more than what single path TCP will take on a path or any collection of paths.[7]

B. Analysis of MPTCP messages.

As a baseline analysis, we measure the performance of MPTCP compared to SPTCP. Then, we analyze the problems of using MPTCP under unbalanced traffic conditions in the network.
Figure 6 shows our testbed setup. MPTCP v0.90 is installed on Ubuntu 14.04 for both client and Web server.The client uses two wireless adapters to access 802.11g and 802.11n simultaneously. The WiFi access point has three dual band antennas and can provideup to 300 Mb/s over 802.11g and up to 450 Mb/s over 802.11n theoretically. The signal strength measured at the client's device is -45 dBm over 802.11g and -38 dBm over 802.11n. The RTT between the client and the server is less than 10 ms.Our TCP tuning is as follows: Caching slow start threshold (ssthresh) from the previous connection is

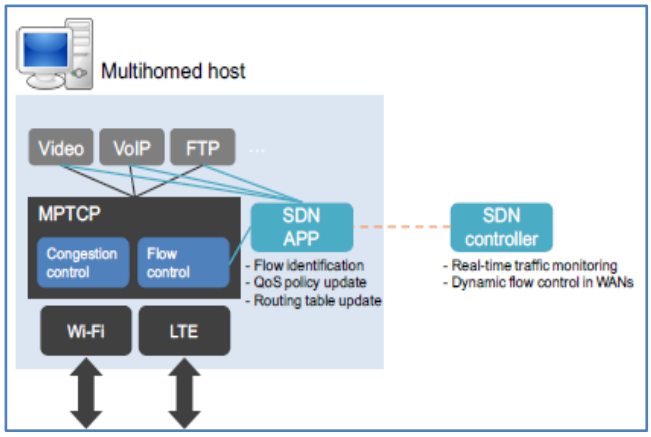
disabled. The initial congestion window size (CWND) is 10 and TCP Selective Acknowledgment (SACK) is enabled. The server's send buffersize (wmem) settings are 10KB (min.), 85KB (default) and 16MB (max.) and the client's receive buffer size (rmem) settings are



4KB (min.), 85KB (default) and 6MB (max.).[7]
Fig.5. Testbed setup for a baseline analysis of MPTCP

C. Measuring MPTCP performance

To avoid the fairness issue of using MPTCP, there is only a single client operating either through MPTCP or SPTCP during each experiment. We compare the download complete times while the client is downloading several files of various sizes (10KB through 5 MB) from the Web server. The number of data samples for each set of experiments associated with a given file size is 100. For a small file size such as 10 KB, we do not find a significant difference between MPTCP and SPTCP. This is because before establishing the second subflow over 802.11g,the client completely downloaded the entire file via 802.11n. For 100KB file size, MPTCP shows a slightly faster download than SPTCP. The client received the majority of the data during the slow start over both paths. For a large file size such as 1MB and 5 MB, MPTCP clearly outperforms SPTCPby aggregating the bandwidth over two paths simultaneously during a download.
The performance also affects how fast MPTCP establishes a second flow. During the above experiments, it took less than 50 ms to send the SYN segment over 802.11g for the second flow after the first subflow was established over 802.11n. The other question is, how quickly does MPTCP adapt to sudden changes in network conditions such as when an existing path gets disconnected or a new interface becomes available during the file transfer. For this experiment, we repeatedly connectedand disconnected the 802.11g adapter in the middle of a download. We first disconnected the 802.11g adapter and tracked the sequence numbers of the MPTCP in-flight packetsthat were sent over 802.11g but not acknowledged yet. It took 0.9 s until those packets were resent over 802.11n. Then, we connected the 802.11g adapter back again, and observed thatMPTCP sent the SYN segment to re-establish the second flow as soon as the client was re-connected to the 802.11g



(within less than 1 s).
Fig. 6. Dynamic MPTCP path control using SDN

D. DYNAMIC MPTCP PATH CONTROL USING SDN

We show that MPTCP may experience poor performance over multiple paths that have significant differences in bandwidth availability. Based on the analysis, we suggest to dynamically adjust the number of MPTCP paths with the support of SDN. The question is, how do we recognize such heavily unbalanced traffic load conditions? And, when and how do we adjust the number of MPTCP paths during a download? Figure 7 shows our proposed platform. An SDN application running on the user's device retrieves various networking information from an SDNcontroller in the WAN and selects appropriate MPTCP paths in real time. To avoid scalability issues, the MPTCP client can obtain the information directly from the SDN-enabled local edge nodes such as Wi-Fi access points or eNodeBs in Figure 7.[7]
An SDN technology is designed to be deployed among switches in data centers, and its availability has been extended to routers in WANs. Using an OpenFlow protocol, an SDN controller obtains various networking feedback (e.g., current capacity and packet loss rate on a link) from WAN routers

in real time. Then, the controller dynamically changes routing paths based on changing network conditions and QoS rules that can be updated by service providers using SDN Northbound APIs.

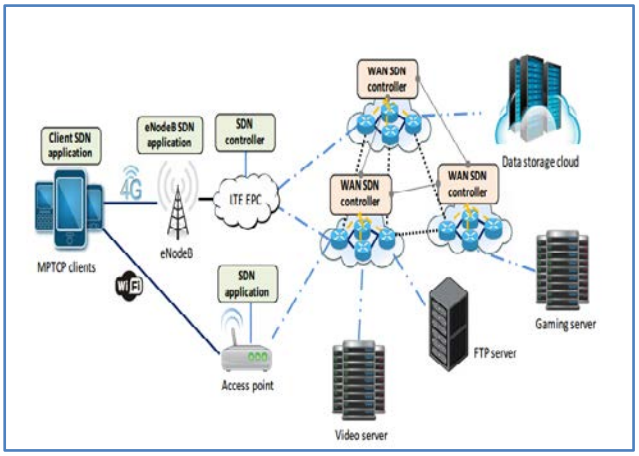


Fig.7. Proposed SDN platform in WANs

3 On the Benefits of Using Multipath TCP and Openflow.

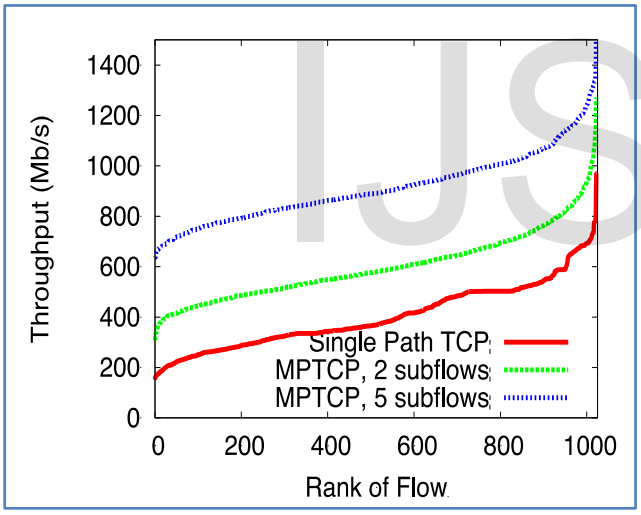


Fig.8. the benefits of using MTCP & Openflow

3.1 BackboneNetworks and Access Networks.

- Backbone networks.

- Abundant and redundant network resources (links and routers/switches) for large bandwidth and high reliability.
- A part of backbone networks may continue to survive even if a large scale link/node failure occurs due to a large disaster.
- Utilizing still available network resources (links, nodes) could enable the network to continue providing acceptable services.

- Access networks.

- located close to users, usually not redundant: once a disaster breaks down access networks, it may be very difficult to quickly repair them.
- Rather than repairing the destructed access network, network users in the disaster area could construct their access network, using any available devices more quickly and more easily.

A. Requirements to Resilient Backbone Networks.

- Network resilience could be measured by the Network Recovery Time with two major time components:

  a. Failure Detection Time: detection of alarms and alerts to locate network faults.

  b. Switchover Time: disables a failed port, enables another, reroutes traffic around a failed switch or router.

- For failure detection, existing detection technologies like BFD (Bidirectional Forwarding Detection) could be utilized.
- For switchover, we apply SDN (Software Defined Networking)
- /OpenFlow technology because SDN/OpenFlow has a potential capability to provide more programmability and flexibility to respond faster to network situational changes than existing technologies (like MPLS).
- For Network Recovery Time, at most 50 ms is considered tolerable to complete path restoration, in the provider networks.[10]

3.2 SDN/OpenFlow for Backbone Networks

SDN/OpenFlow provides an easier way to manage and automate networks by separating the control plane and the data plane.

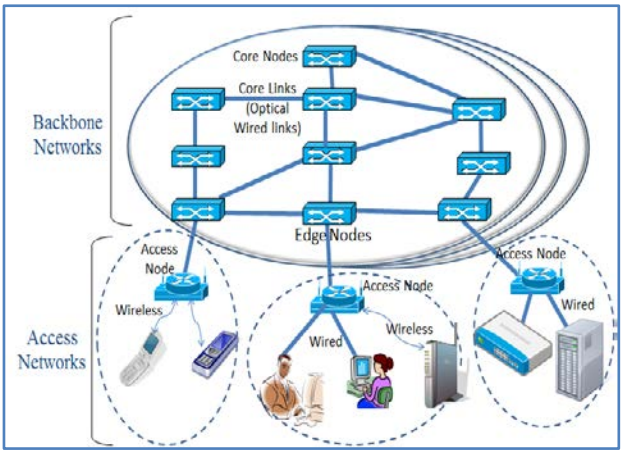


Fig.9. Backbone Network and Access Networks

3.3 Goal of Resilient Backbone Network

To provide non stoppable end-to-end services in the various critical environments, including link/path and node failures.

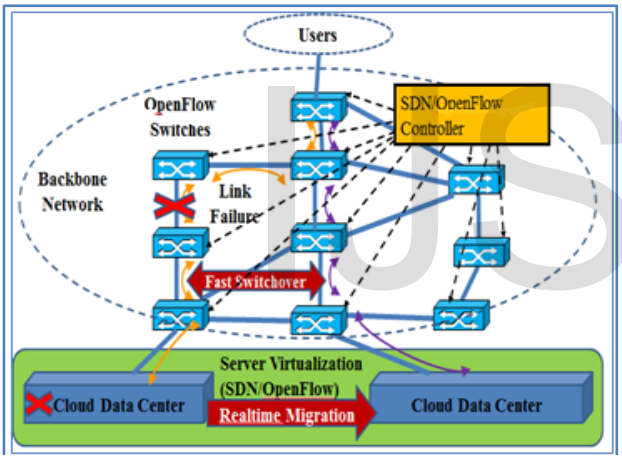


Fig.10. MTCP & OpenFlow cohabitation

There are three issues for Fast network recovery using SDN/OpenFlow technology as noted below:

1. Switchover mechanism from a faulty link to a normal link

   - Switchover time: the time from failure detection to path restoration on an end-to-end basis.
   - OpenFlow: a wide variety of switchover mechanisms.
   - Implementation of several OpenFlow–specific and OpenFlow-integrated switchover mechanisms and evaluations of switchover performance.
   - OpenFlow–specific switchover mechanisms:

- FAILOVER GROUP TABLE-based implementations
- SELECT GROUP TABLE-based implementations
- Both utilize local states of OpenFlow switches without direct involvement of remotely located controllers.
- OpenFlow-integrated switchover mechanisms: OpenFlow with (MTCP) in

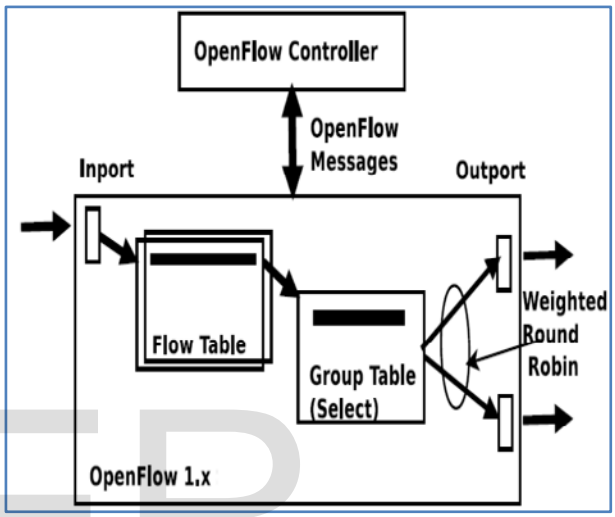


Fig.11. OpenFlow–specific switchover mechanisms

3.4 Global view of the network: correct information on topology and link status

- Necessary to find any available network resources (paths, links) to restore all the end-to-end paths.
- IP network: a global view is maintained by the IP routing protocols like OSPF and BGP, but slow convergence time is a problem.
- SDN/OpenFlow network:
  - a global view could be

maintained among multiple SDN controllers, by existing IP routing protocols or any new routing protocols for SDN/OpenFlow) but there are very few standardized routing protocols especially designed for SDN/OpenFlow.

A. FAST FAILOVER GROUP TABLE

fast failover group tableallows a fast switchover from the active output port to the standby output port. (in the active/standby mode) without direct involvement of controller.[7]

B. SELECT GROUP TABLE.

The functioning of this mechanism was based on following steps:

- SELECT GROUP TABLE allows a single data flow to be divided into multiple subflows, each with a different path (output port) in a weighted round robin manner (in the active/active mode)
- When link/port failures occur, the switch recalculates the weighted values, eliminates the failed ports and reallocates the traffic to active output ports.
- SELECT GROUP TABLE achieves a better resource allocation and less packet loss than FAST FAILOVER GROUP TABLE.

C. Multipath TCP (MPTCP) Integrated with OpenFlow.

- MPTCP is standardized by IETF (Internet Engineering Task Force), does not need to modify existing applications.
- MPTCP creates and maintains multiple active paths for an end–to-end connection.
- Divides the TCP flow into multiple active TCP subflows.
- Each subflow may go through a different path to achieve better resilience.OpenFlow achieves fast switchover among multiple active paths when some of the paths fail.

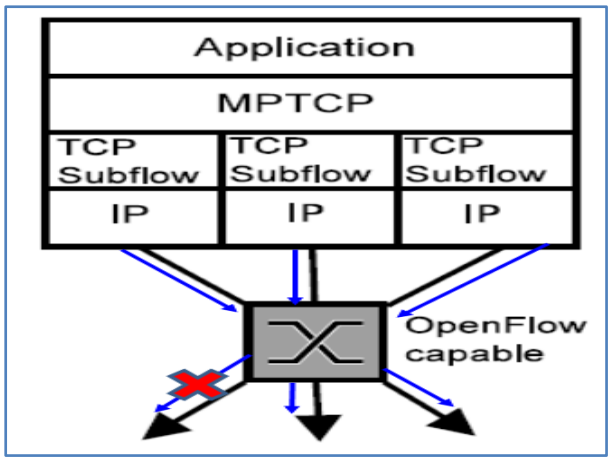


Fig.12. fast failover group table mechanism

4 Implementation and Evaluation of MPTCP on SDN Environment

Two subflows each with a different path through a different WiFi access point .When the path1 fails, path2 keeps transferring the added path1 traffic to achieve seamless handover from Path1 to Path2.

4.1 Communication Delay (Latency) between Controllers and Switches

In the OpenFlow framework, packet forwarding (data plane) and routing decisions (control plane) run on different devices. OpenFlow switches are in charge of packet forwarding, whereas a Controller, which can be situated very far from a networking point of view from the switches its manages, sets up switch forwarding tables on a per-flow basis. The connection between a switch and its Controller is thus of primary importance for the performances of the network.

4.2 Global View of the Network.

- Mininet 2.0, the virtual network simulator is used to investigate the overall behaviors of link network recovery under the Great East Japan Earthquake scenario for SINET3 topology.
- The worst case latencies between the controller and switches are assumed
- When a link failure occurs on the main path, the controller software (POX) with the global view should install new rules to the switches to switchover the traffic flow from the faulty path to the backup path.

4.3 Network Recovery Simulation Result

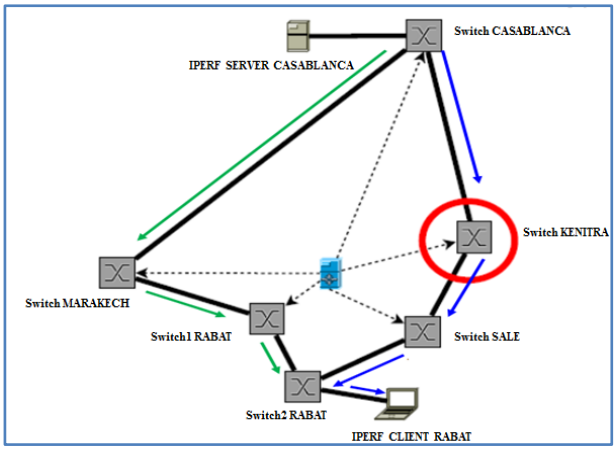


Fig.13. *Global View of the Network.*

When a link failure occurs at 10th and 50th seconds, we confirmed that the traffic flow is effectively turned from the faulty path to a new path thanks to the POX controller's global view of the network.

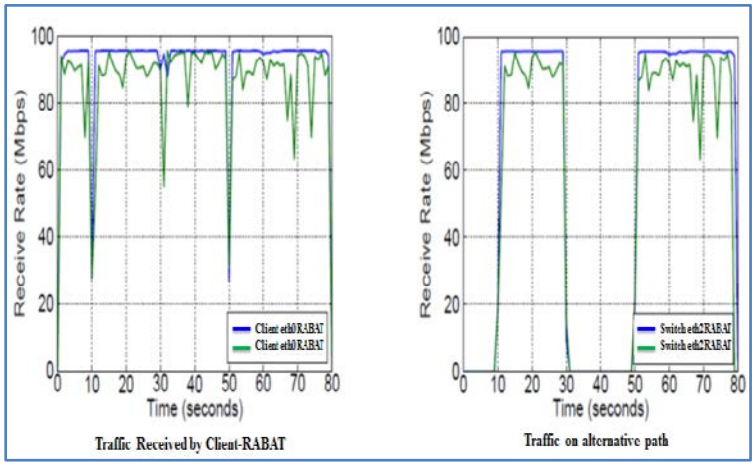


Fig.14. Simulation Result

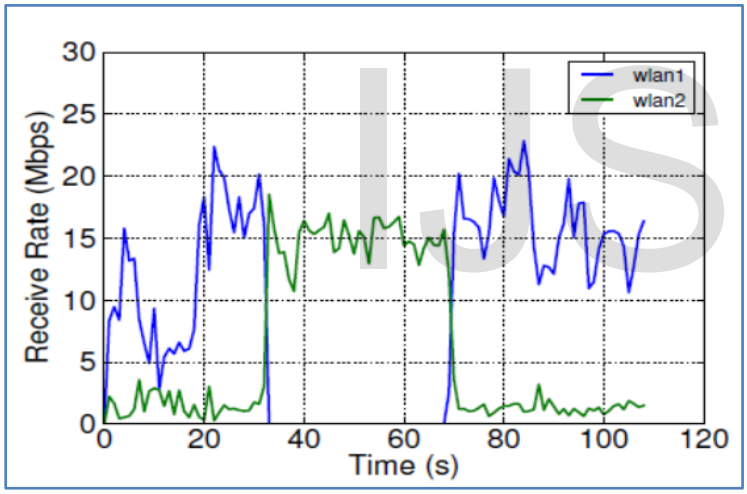


Fig.15. Simulation Result

CONCLUSION

We show how MPTCP performance can be improved in an SDN-enabled network. In our proposed platform, MPTCP can dynamically control subflows based on the available capacity of connected paths with the support of SDN controllers. We will evaluate our proposed solution using Mininet over WiFi networks, and demonstrate the benefit of an algorithmic framework that dynamically enables back and forth switching between best available single paths (single path TCP) and multipath (MPTCP), with support of a SDN controller, in order to maximize downloading rates.

REFERENCES

[1] W. Stallings, "Software Defined Networks and OpenFlow," in The Internet Protocol Journal, vol. 16, no. 1, March 2013, pp. 2-14.

[2] Transmission control protocol. http://www.ietf. org/rfc/rfc793.txt.

[3] O. S. Consortium et al., "OpenFlow Switch Specification v.1.3.4," March 2014 pp. 1-171

[4] Lorena Isabel Barona , "Extending OpenFlow in Virtual Networks" March 2015 pp. 252-258.

[5] M. Suh, S. H. Park, B. Lee, and S. Yang, "Building firewall over the software-defined network controller," in Advanced Communication Technology (ICACT), 2014 16th International Conference on, 2014, pp. 744–748.

[6] "Home - Open Networking Foundation." [Online]. Available: https://www.opennetworking.org/. [Accessed: 07-Jun-2014].

[7] T. D. Nadeau and K. Gray, SDN: Software Defined Networks, 1 edition. O'Reilly Media, 2013. [9] "Software-Defined Networking: The New Norm for Networks." ONF, 13-Apr-2012.