Sep. 11, 2016
Haosen Wen
NSF Miniproposal

Title: DPI-Enhanced SDN Networking Based on Openflow™
NSF Organization Unit: CNS – Division of Computer and Network Systems
Program: NeTS – Networking Technology and Systems

Project Summary

Overview:

In various computer networks, SDN provides a control plane separated from data plane, through which engineers and researchers may implant new algorithms, especially global ones, for optimized traffic engineering strategies either under research or industrial application.

Recently, datacenter networks (DCNs) are the frontier of applications of SDN. Typical traffic engineering algorithms of DCN upon SDN try to set optimized route (or routes) and forwarding strategies for the data stream of one or more kinds of applications that shows similar features during transmissions. For example, data streams of video or audio stream usually requires a steady, non-stop stream, and/or even a stream with lower out-of-order rate than expected.

Restrained by current SDN protocol, such as Openflow™, switches in an SDN network can only get and transmit flow information up to IP layer, which makes it difficult for developers to identify the type of a certain data flow – they can only guess the type of flows according to their patterns, which is vague since the target type of flows don' t always follow a fixed pattern, and vice versa.

And that' s where deep packet inspection (DPI) comes in. With DPI, network devices (like SDN switches) are able to analyze the type of each packet up to application layer, and give more precise information of every flow in the network to the controller, or apply the flow instructions to those flows of the exact type of application, which can make the flow control much more efficient and accurate.

Potential Benefits:

The direct benefit that DPI can provide to present SDN networks is to fully recognize the protocol of every dataflow in the networks, which might lead to the following useful applications:

Enhanced Network Security: Attacks like DDoS have become a major problem in network security area. Still, DPI may not be able to prevent DDoS from happening, but with a quick recognition of suspicious protocol or flow pattern down at networking layer, it would be faster to identify and stop a DDoS attack with pre-installed strategy in controller. Tommy Chin(2015) mentioned such a defending mechanism in SDN networking based on DPI in his work.

Accurate Flow Classification & Routing: Many present routing strategies are designed for certain kinds of application-layer protocols, which may sometimes aim at wrong flows due to limited flow information. With DPI, those strategies can be led precisely to the desired dataflows.

Brief Description of Research:

By now, the most widely-used SDN protocol is Openflow™, which defined all required specification of an Openflow™ switch and thus the communication standards between the switch and an SDN controller. So making an SDN switch able to perform a deeper package inspection is in fact an extension of Openflow™ protocol.

To fulfill this task, research and experiments upon both data plane and control plane must be done.

Data plane:

For data plane, the program will start from researches upon a popular open-sourced virtual SDN switch, OpenvSwitch(OVS). The key problem of implanting a DPI module into OVS would be the optimization of SDN algorithms. OVS decodes and analyze all the packet received through an express channel built in Linux kernel, which ensures that the switch won't have to run an upcall process from kernel mode to userspace every time a packet comes, so that it won't delay the transmission and forwarding of data packets. However, DPI, as an advanced kind analyzation of single packets or flows, might take much more calculation than decoding up to IP layer only, since DPI process contains not only decoding of certain keywords of flows, but also content pattern recognition, key word recognition and flow pattern recognition, which requires much more complicated algorithms than traditional process. Hence, this process requires refined packet sampling algorithms to reduce the cost analyzing packets, and to ensure the accuracy of DPI results at the same time.

What's more, the architecture of the DPI module in OVS will also greatly influence the performance of the system, since the whole DPI process may contain multiple algorithms of two or more layers. Some of the algorithms are simple, quick-to-run but can only give analysis results of some kinds of flows that have distinct features, while others may take longer time, yet also could be run less frequently. Take an open-source DPI library, nDPI, as an example, it will firstly guess the protocol of a dataflow by finding a match from a list of IP addresses and ports of known hosts. If the guess failed, it would launch its pattern recognition modules, which is much more complicated. With a good architecture of DPI module, some frequently used simple algorithms are implanted down to kernel level to achieve better performance of DPI without causing considerable delay.

Control Plane:

Given that flow control based on DPI requires flow entries with information of flow protocols, one or more example protocol-oriented traffic engineering strategies should be

given to test the overall performance of the enhanced network. In fact, plenty of flow-oriented strategies under research are logically protocol-oriented – they are just trying to get the target flows of certain protocols by guessing with simple IP-layer patterns, since hardly any further information can be given to the controller. In this case, complete strategies wouldn't be necessary.

However, with information set of the communications between switches and controllers different from previous SDN network (which contains protocol information of the flows), proper data structures for these new information should be updated in the protocol as an extension.

Resources Required

SDN Testbed: For an emulation of a SDN datacenter network, a common server will be sufficient. For a testbed consist of physical devices, we need 4~8 common PCs as terminals, a common server with a controller built in, and an Openflow™ switch with more than 8 ports.

Human Resource: A professor (as supervisor of the program), and a PhD student with SDN and Linux kernel programming background, and a graduate student that familiar with networking programming and SDN switch architecture.