

氏名	Phan Xuan Thien
学位(専攻分野)	博士(情報学)
学位記番号	総研大甲第 2116 号
学位授与の日付	2019 年 9 月 27 日
学位授与の要件	複合科学研究科 情報学専攻 学位規則第6条第1項該当
学位論文題目	Fine-Grained and Distributed Traffic Monitoring Platform in Software-Defined Networks
論文審査委員	主査 准教授 福田 健介 教授 計 宇生 准教授 栗本 崇 准教授 鯉淵 道紘 准教授 関谷 勇司 東京大学情報基盤センター

(Form 3)

Summary of Doctoral Thesis

Name in full Phan Xuan Thien

Title

Fine-Grained and Distributed Traffic Monitoring Platform in Software-Defined Networks

Traffic engineering is an important issue for network operation. It is essential for most networks since it enables network operators and service providers to ensure efficient use of network resources and proper network performance for applications and services. Traffic engineering adapts the routing of traffic based on the network conditions and optimizes traffic demand and capacity such as big flow migrations, fine-grained QoS control, anomaly elimination. Therefore, it requires integrating of traffic monitoring and control capabilities in the network. However, in the past and current networks, network monitoring and control are conducted independently, and current monitoring techniques require separate hardware deployment or software configuration, making it hard to implement traffic engineering and other network management applications.

Different monitoring techniques are used for monitoring networks such as sFlow, NetFlow, Simple Network Management Protocol (SNMP), and other telemetry tools. Though SNMP is integrated in most network devices, it limits counters to aggregate traffic for the whole switch and each of its interfaces, disabling insight into flow-level statistics necessary for fine-grained traffic engineering. While packet-sampling tools like sFlow and NetFlow mostly require separate hardware deployment for the flow collector and they are not integrated with existing control protocol/APIs in the networks. Therefore, existing monitoring techniques remain inflexibility and drawback to meet traffic engineering requirement.

Recently, Software Defined Network (SDN) has been introduced to solve the drawbacks of network control and monitoring in current networks. SDN in general and OpenFlow as its current implementation instance in particular, provides a centralized visibility with global network and application information, a programmability without a need to handle individual network elements, and a traffic flow based controllability with flow table pipelines in OpenFlow switches making flow management more flexible and efficient. With these supports, traffic engineering mechanisms can be implemented flexibly and intelligently in OpenFlow compared to

conventional approaches.

For traffic monitoring functionality, OpenFlow employs a default monitoring mechanism that records statistics of flows using forwarding flow tables. This monitoring mechanism may not ensure an effective performance for fine-grained monitoring (i.e., monitoring network traffic with high granularity) due to two main reasons: (1) installing larger number of fine-grained rules without coarse-grained forwarding rules increases cache misses, which results in additional latency that decreases the performance of the switch as it may send a Packet-In message to the controller to ask for a forwarding rule and wait for controller's response; (2) installing larger number of flow entries make the sizes of flow tables greater, which increases the latency due to the lookup process for matching incoming packets with the flow entries. Therefore, an effective monitoring method with better performance at switch is critically essential to improve traffic monitoring in OpenFlow for fine-grained traffic engineering.

In addition, in most OpenFlow based networks, a number of selected OpenFlow switches for monitoring tasks independently monitor traffic flows. These switches may consume huge resources (e.g., throughput, CPU, memory usage) to perform monitoring tasks. When monitoring a network in a distributed scenario, for each flow that traverses through multiple monitoring switches, the switch along the flow path records almost the same statistics of the flow. This introduces a duplication issue of flow monitoring as a single switch in the flow path is enough to monitor the flow statistics. This duplication results in redundant flow-based monitoring rules in switches that consume significant resources of the switches and the network, and may cause serious problem to the performance of the switches and the network due to their limited resources. Therefore, a distributed monitoring capability that can distribute the monitoring load over multiple monitoring switches in the network and eliminate duplication is critically essential for monitoring in distributed scenarios.

In this dissertation, we propose a systematic approach that integrates fine-grained traffic monitoring capability to a capable traffic control platform, i.e., OpenFlow, for traffic engineering applications. Our approach concentrates on both solving the monitoring performance limitation at OpenFlow switch (i.e., OpenFlow software switch in particular), and enabling a distributed monitoring capability at controller for flexible and low overhead flows monitoring that operates independently from the forwarding functionality in the switch. In the proposed method, network flows statistics are actively monitored based on monitoring match fields (e.g., 5-tuple match fields) that can be defined by controller applications. Traffic flows are forwarded based on flow entries of flow tables while their statistics at a fine-grained level are monitored at a monitoring module that is independent from the forwarding tables. The approach ensures network flows are monitored at switch with low overhead

independent of the forwarding functionality of the switch, and applications use such flow data statistics via extended OpenFlow APIs. As a result, the proposal decreases the monitoring overhead in the switch even for monitoring large number of active flows.

Furthermore, for distributed monitoring scenarios, we propose a distributed monitoring method that eliminates the redundant monitoring rules, and distributes the monitoring load over multiple monitoring switches in a balancing fashion. The proposed method detects duplicated monitoring rules and for each duplication, it selects a switch with highest availability among the switches along the path to monitor the flow and eliminates the redundant monitoring rules in the other switches in a balancing fashion. The switch selection is adaptive based on the available status of the switches, which is frequently updated in each statistics query time of the controller. The proposed method decreases the number of monitoring rules per switch, therefore it decreases the monitoring load of the switches and the entire network.

We implemented our proposed methods as a monitoring platform integrated to OpenFlow called SDN-MON. We also designed and implemented a dedicated protocol for the communication between the switches and the controller for exchanging monitoring control messages and transmitting the monitoring data. The designed protocol is implemented with OpenFlow based formats to integrate it to OpenFlow standard. We conducted a number of experiments based on the implementation instance to show the effectiveness of our proposals. The experimental results demonstrate a low monitoring overhead at switch, and a low processing time of the proposed distributed monitoring mechanism at controller.

博士論文審査結果

Name in Full
氏名 Phan Xuan Thien

Title
論文題目 Fine-Grained and Distributed Traffic Monitoring Platform in Software-Defined Networks

本論文は、ソフトウェア・デファインド・ネットワーク(SDN)における、トラフィックエンジニアリングのためのトラフィックフローのモニタリングフレームワークに関して行った申請者の研究内容をまとめたものである。既存のSDNでは、トラフィックエンジニアリングに欠かせないトラフィックの柔軟な制御機構が提供されているものの、その制御を行うに必要なスイッチおよびコントローラでのトラフィックフロー情報の収集については適切なサポートが行われていない。とりわけ、モニタリングによるスイッチでのトラフィック転送オーバーヘッド、および、複数スイッチが独立にモニタリングを行うことによる非効率的な資源利用によるコストが問題となる。これらの問題をSDNの代表的なフレームワークであるOpenflowを用いたソフトウェアスイッチ(Lagopus)、コントローラ(Ryu)において解決する手法を示すことが本研究の目的である。

論文は6つの章から構成される。第1章では研究の背景と目的、第2章では既存のネットワークモニタリング手法、SDNにおけるネットワークモニタリング手法や、ターゲットとなるアプリケーション・ネットワークにおけるフロー数の規模に関する関連研究について述べている。

続いて、第3章では単一スイッチ内でのフローモニタリングフレームワークに関するデザイン・実装・評価について述べている。従来手法ではフローテーブルをスイッチングおよびフローモニタリングに使用することに対して、提案手法では、フローテーブルを2つに分け、独立したモニタリング用のテーブルを用意し5タプルフローのモニタリングに特化することで、フローレベルの効率的なモニタリングをOpenflowの拡張として実現している。Lagopus上に実装したフレームワークは、フロー数が多い場合にも転送性能の劣化は限定的であり、既存手法に対して効率的にフローモニタリングが可能であることを示した。

第4章では、個々のスイッチから集められたフロー情報を効率的に処理するためのコントローラでの拡張について述べている。従来の手法では、個々のスイッチが独立にフローをモニタリングするため、コントローラでは同じフローを異なるスイッチで複数回カウントすることが起こりうる。また、ネットワークトポロジに応じて、各スイッチのモニタするフロー数に偏りが生じるため、スイッチのモニタリングによる負荷が不均一となる。この問題に対して、各スイッチでモニタしたフローをコントローラに集約後、コントローラにてフローの重複検出・削除、およびスイッチあたりのフロー数の負荷分散を行うフレームワークをデザイン・実装・評価している。提案フレームワークは、ソフトウェアコントローラ(Ryu)上に実装され、前述のスイッチの拡張と合わせて、システム全体としての評価

を行った。複数台のスイッチを用いた評価では、正しくフロー数がスイッチ間で分散されること、想定したアプリケーションが使用するために十分な時間間隔での動作が可能であることを示した。

第5章では提案フレームワークの性能・制限、アプリケーション例について述べ、最後に第6章では結論と今後の研究課題・展望を示した。

なお、研究成果として、申請者は主著で査読付ジャーナル 1 篇、査読付国際会議論文 1 篇、査読付国際会議論文(ショート)1 篇の発表を行っている。

以上を要するに、本論文は SDN におけるトラフィックエンジニアリングに必要なフローモニタリングに関する課題として、各スイッチでのモニタリングオーバーヘッドの削減とスイッチ間での冗長なフロー情報の削減・負荷の均一化をはかるフレームワークを設計・実装・評価し、その有効性を示したものである。以上の理由により、審査委員会は、本論文が学位の授与に値すると判断した。