L-012

OpenFlow を用いたレイヤ 7 ペイロードの省略による低負荷なパケットミラーリングシステムの実装

Implementation of Low-Load Packet Mirroring System by Omitting Layer-7 Payload Using OpenFlow

清水 貴弘 † 山井 成良 ‡ 北川 直哉 ‡
Takahiro Shimizu Nariyoshi Yamai Naoya Kitagawa

1. はじめに

ポートミラーリングでは，その特性上，双方向のトラフィックが 1 つのミラーポートに集中することになる．そのため，ポートミラーリングを実施している時に，ミラーデータの流量がミラーポートの帯域を超えてしまうと，すべてのパケットのミラーリングが行えなくなる．

本論文では，OpenFlow を用いて，大容量の転送を行う通信において，通信内容の解析に必要な通信の一部のパケットのみをミラーリングし，大部分のレイヤ 7 ペイロードのパケットのミラーリングを省略することで，負荷の低いミラーリングを実現するシステムについて述べる．

2. 一般的なポートミラーリングの問題点

ポートミラーリングとは，スイッチやルータが持つ機能の一つで，あるポートが受信したパケットを同時に他のポートに送信する機能である．一般的なポートミラーリングでは，あるポートで受信したすべてのパケットについてミラーリングを行うため，ミラーポートの帯域が不足することがある．この問題を緩和するため，ACL(Access Control List) フィルタリングを用いて，特定の条件に合致するパケットのみを選択的にミラーリングする機能を持ったネットワークスイッチが存在するほか，OpenFlow でも実現可能である [1]．しかし，パケットフィルタリングの条件がスイッチを通過するパケットの大部分を占める場合，一般的なポートミラーリングと同様にミラーポートの帯域が不足する恐れがある．

3.OpenFlow を用いた省略ミラーリングシステム

3.1. 実現方針

本論文では OpenFlow を用いてパケットの内容を調べ，以降のパケットの解析に必要なレイヤ 7 のヘッダ情報の取得が終わった時点でミラーリングを停止することにより，ミラーポートの帯域の利用を大幅に減らし，パケットキャプチャ後の解析に必要なパケットの消失を防ぐシステムを提案する．

本システムでは，OpenFlow コントローラは通信の序盤のみにパケットの情報を調査する処理を行うが，それ以外では一般的なレイヤ 2 スイッチとして動作する．OpenFlow コントローラで解析に必要な情報が含まれていることを確認できれば，その以降の通信内容は省略可能であるため，ミラーリングを実施せずにスイッチングを実施する．これにより，帯域を圧迫するトラフィックを大幅に削減することが可能となる．

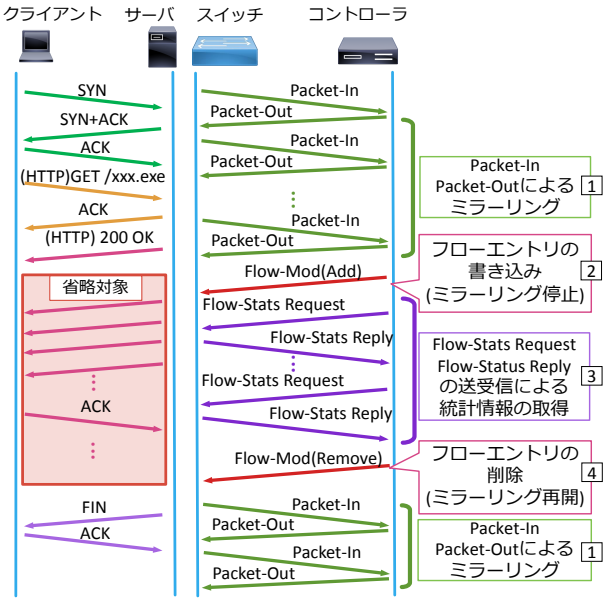


図 1: HTTP 通信を例としたシステムの通信の様子

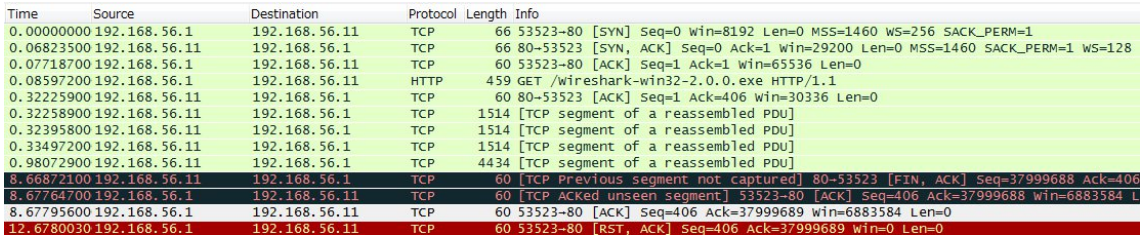
3.2. 提案システムの動作

本節では，提案システムの動作について述べる．なお，本システムの動作例として，大容量の HTTP 通信を用いたファイル転送時の動作について示す．図 1 は動作中の本システムにおける，クライアントとサーバとの間の通信および OpenFlow スイッチと OpenFlow コントローラとの間の通信の流れを示す．本システムは以下に示す流れで動作する．なお，図 1 中の番号と以下の番号はそれぞれ対応している．

(1) パケットに含まれるレイヤ 7 の情報を OpenFlow コントローラで解析して，パケットの解析に必要なパラメータを探す．パラメータが見つからない間は，ミラーリングを実施し続ける．

(2) OpenFlow コントローラがパラメータを発見したら，OpenFlow スイッチにフローエントリを書き込んで，パケットのミラーリングを停止する．

†東京農工大学工学部情報工学科，Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology
‡東京農工大学大学院工学研究院先端情報科学部門，Division of Advanced Information Technology & Computer Science, Institute of Engineering, Tokyo University of Agriculture and Technology



Time	Source	Destination	Protocol	Length	Info
0.00000000	192.168.56.1	192.168.56.11	TCP	66	53523→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
0.06823500	192.168.56.11	192.168.56.1	TCP	66	80→53523 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
0.07718700	192.168.56.1	192.168.56.11	TCP	60	53523→80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
0.08597200	192.168.56.1	192.168.56.11	HTTP	459	GET /Wireshark-win32-2.0.0.exe HTTP/1.1
0.32225900	192.168.56.11	192.168.56.1	TCP	60	80→53523 [ACK] Seq=1 Ack=406 Win=30336 Len=0
0.32258900	192.168.56.11	192.168.56.1	TCP	1514	[TCP segment of a reassembled PDU]
0.32395800	192.168.56.11	192.168.56.1	TCP	1514	[TCP segment of a reassembled PDU]
0.33497200	192.168.56.11	192.168.56.1	TCP	1514	[TCP segment of a reassembled PDU]
0.98072900	192.168.56.11	192.168.56.1	TCP	4434	[TCP segment of a reassembled PDU]
8.66872100	192.168.56.11	192.168.56.1	TCP	60	[TCP Previous segment not captured] 80→53523 [FIN, ACK] Seq=37999688 Ack=406
8.67764700	192.168.56.1	192.168.56.11	TCP	60	[TCP ACKed unseen segment] 53523→80 [ACK] Seq=406 Ack=37999688 Win=6883584 L
8.67795600	192.168.56.1	192.168.56.11	TCP	60	53523→80 [ACK] Seq=406 Ack=37999689 Win=6883584 Len=0
12.6780030	192.168.56.1	192.168.56.11	TCP	60	53523→80 [RST, ACK] Seq=406 Ack=37999689 Win=0 Len=0

図 2: 省略ありの場合のパケットキャプチャ結果

(3) パケットのミラーリングを停止している間，OpenFlow スイッチを通過したパケットの量をフローエントリの統計情報を用いて取得し，通信の終了時間を推定する．

(4) 推定した終了時間を基に，フローエントリを削除し，ミラーリングを再開させる．以降，(1) から (4) の動作を繰り返す．

3.3. ミラーリングパケットの削減効果の検証

動作例として，クライアントがサーバから約 36MB のバイナリファイルを HTTP 通信の GET リクエストによりダウンロードする通信を発生させた場合について，削減効果の検証を行った結果を示す．本検証では，Open vSwitch にサーバ，クライアント，監視端末に見立てた仮想マシンを複数用意し，Open vSwitch を OpenFlow スイッチとして，Trema を用いて作成した OpenFlow コントローラで通信の制御を行う．仮想環境のハイパーバイザとして Oracle VirtualBox 5.0.2 を使用し，ホストマシンに Intel Core i7-4500U(1.80GHz)，メモリ 8GB の Windows8.1 を搭載したコンピュータを使用した．

図 2 に，パケットキャプチャツールである Wireshark を用いた監視端末におけるパケットキャプチャの結果を示す．なお，結果はこの検証において省略対象とした HTTP 通信についてのパケットのみを示した．また，表 1 に OpenFlow を用いて実装した通常の L2 スイッチによるポートミラーリングの結果との比較を示す．

図 2 は，本システムを用いてパケットの省略を実施した時のパケットキャプチャの結果であり，対象とした HTTP 通信について受信した 13 パケットすべてを示している．図 2 中において，TCP 通信におけるシーケンス番号と確認応答番号が合わないために一部のパケットについて警告が出力されている．これは，本システムによって TCP 通信の一部が省略されたため，当該パケットが観測されなかったことを示し，本システムが正常に動作したことを表している．

約 36MB のファイルの転送において，本システムによる省略後のミラーリングパケットのデータ量は表 1 に示すように 9,927 バイトであり，約 35MB の削減を観測した．また，本システムがミラーリングする対象とするのは HTTP 通信のヘッダ部分などの L7 ヘッダのみであるため，さらに大きなファイルの送受信の場合にはミラーリングパケットの削減量はより大きくなり，さらなる効果が期待できる．

表 1: システムの動作結果

	パケット数 [個]	データ量 [Byte]
省略なし	7,418	10,444,377
省略あり	13	9,927

4. 考察

本システムは，ネットワーク型の IDS(Intrusion Detection System) のような，継続してパケットを収集するシステムと相性が良いと考えられる．ミラーリングパケットの量の現象による負荷軽減効果が期待できるほか，通信のヘッダ部分が省略されずにミラーリングされることからも，通信内容の解析の精度向上が期待できる．さらに，本システムの付加機能として，TCP FIN パケットや TCP RST パケットを受信した際に即座にミラーリングを再開させる機能を追加することで，安定性の向上が期待できるほか，常時接続を行わない通信方式であれば，Flow-Stats Request/Reply による通信終了の推定を行う必要はないと考えられる．

5. おわりに

本論文では，ポートミラーリングにおける帯域不足を解消するため，通信のうち，レイヤ 7 のヘッダ部分のみをミラーリングし，大部分のパケットのミラーリングを省略することにより，ミラーポートの帯域を圧迫するトラフィックを大幅に削減するシステムを提案した．また，HTTP 通信を対象としたシステムを実装し，通信の一部のみを対象としたミラーリングが正常に機能し，大幅なミラーリングパケットサイズの削減が可能であることを確認した．

OpenFlow コントローラと OpenFlow スイッチ間の通信に遅延が生じる可能性があるほか，OpenFlow の仕様上，パケットの再送を検知することができない．実際の運用時にはこれらの要因によって，通信終了時間の推定の精度が低下する可能性があるため，実運用における検証を行うことが今後の課題である．

参考文献

[1] あきみち, 宮永直樹, 岩田淳. マスタリング TCP/IP OpenFlow 編, 選択的ポートミラーリング, pp. 130–132. オーム社, 2013.