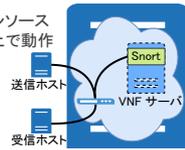


実験環境

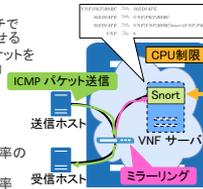
- ◆ NFW を実現するソフトウェア環境の実装を目指したオープンソースプロジェクト **OPNFV** を使用して、1 台の VNF 用仮想サーバと接続用の仮想スイッチを 1 台の物理サーバ内に構築
- ◆ VNF として IDS を実現するオープンソースソフトウェア **Snort** を VNF サーバ上で動作
- ◆ 送受信ホストとして物理サーバを 1 台ずつ用意し、接続



7

実験方法と評価指標

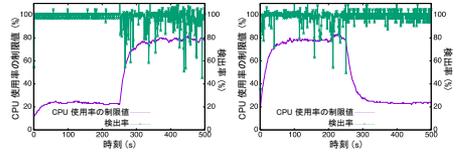
- ◆ 実験方法
 - ◆ 送信ホストから受信ホストへ向けて、設定したレートで ICMP パケットを送信
 - ◆ ICMP パケットが通過する仮想スイッチでミラーリングし、VNF サーバへ到達させる
 - ◆ VNF サーバ上の Snort は ICMP パケットを検出するようにルールを設定して稼動
 - ◆ VNF サーバでは、サービス空間構築手法に基づいて化学反応を実行し、VNF の CPU 使用率を制限
- ◆ 評価指標
 - ◆ Snort のプロセスに対する CPU 使用率の制限値
 - ◆ Snort における ICMP パケットの検出率



8

実験結果

- ◆ 送信する ICMP パケットのフローレートを 250 秒で変化させた時の、CPU 使用率の制限値と検出率を観測



◆ 10 Mbps → 50 Mbps

◆ 50 Mbps → 10 Mbps

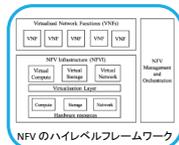
9

まとめと今後の課題

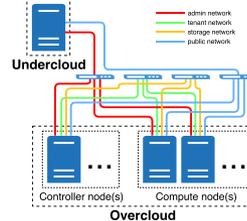
- ◆ まとめ
 - ◆ 生化学反応モデルに基づいたサービス空間構築手法を NFW フレームワーク上に実装
 - ◆ 提案手法に基づいて、VNF への CPU 資源割り当てが適切に行われることを確認
- ◆ 今後の課題
 - ◆ サービスチェイニングの実現
 - ◆ 複数フローの処理の実現
 - ◆ 複数の NFW サーバから構成される大規模環境への拡張

10

NFW フレームワークと OPNFV を用いて構築した NFW 環境



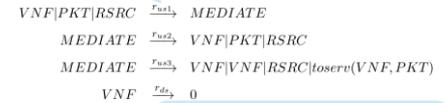
NFW のハイレベルフレームワーク
 VNFs: ソフトウェアで実装されたネットワーク機能
 NFWs: VNF を実行するための様々な資源と仮想化レイヤ
 NFW MANO: 資源と VNF の管理機能とオーケストレーション機能



11

化学反応式

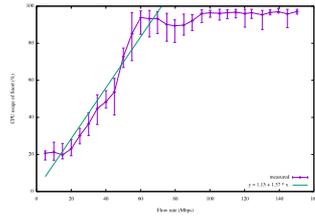
反応速度係数



VNF: VNF 実行の需要
 PKT: VNF が適用されるパケット
 RSRC: サーバで利用可能な資源
 MEDIATE: VNF に与えられる資源
 toserve(VNF, PKT): サービスの実行/結果

12

パケットの送信レートと Snort の CPU 使用率の関係

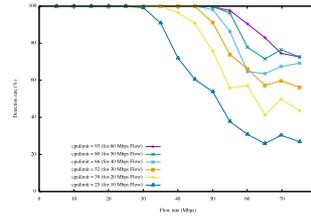


2016/2/21

特別研究助成金 - 松田研究室 杉田崇典

13

手動で CPU 使用率の制限を 行った際の検出率



2016/2/21

特別研究助成金 - 松田研究室 杉田崇典

14