

# Layer-7 フィルタを用いた実 SDN システムにおける マネジメント競合解決手法

中山 裕貴†、林 経正†、阿多 信吾‡  
† 株式会社ボスコ・テクノロジーズ  
‡ 大阪市立大学



# 仮想化技術によるサービスプロバイダの多様化

---

- SDN や NFV を始めとするネットワーク仮想化技術の普及
  - 短期間かつ安価にネットワークインフラを利用可能
    - 運用コストが **50 % 減**
    - 提供までの時間が3週間前後から**数分オーダーに短縮**
  - ユーザの要求に応じた柔軟なネットワーク機能提供可能
    - トラヒックエンジニアリング以外の、**論理的な機能も利用可**
- 物理的および論理的なインフラを借用した上で、独自サービスの提供が容易に
  - 例) 仮想移動体通信事業者 (MVNO)
  - インフラを提供する側：1次プロバイダ
  - インフラを借用し、サービスを展開する側：2次プロバイダ

# 顕著化するマネジメントの競合問題

---

- 1次プロバイダの提供するインフラサービスの変化に伴い、マネジメント要求も変化
  - 1次プロバイダのマネジメントが2次プロバイダのサービスに影響
  - 2次プロバイダの異なる管理運用ポリシーが1次プロバイダのインフラに混在化



- サービスの管理運用が複雑化
- 障害発生の原因究明が困難化

- 円滑なサービス運用のためには、2次プロバイダが主体でマネジメントを行うべき
  - 多量のスイッチやルータ、サーバが存在
  - 混在するベンダやファームウェアのバージョン



運用ポリシーの異なる2次プロバイダに対して  
影響範囲を考慮し、設定することは容易ではない

# 目的

---

- 2次プロバイダを容易に構築可能となったが管理運用が複雑化
  - サービス維持におけるマネジメントプレーンの適切な運用が課題



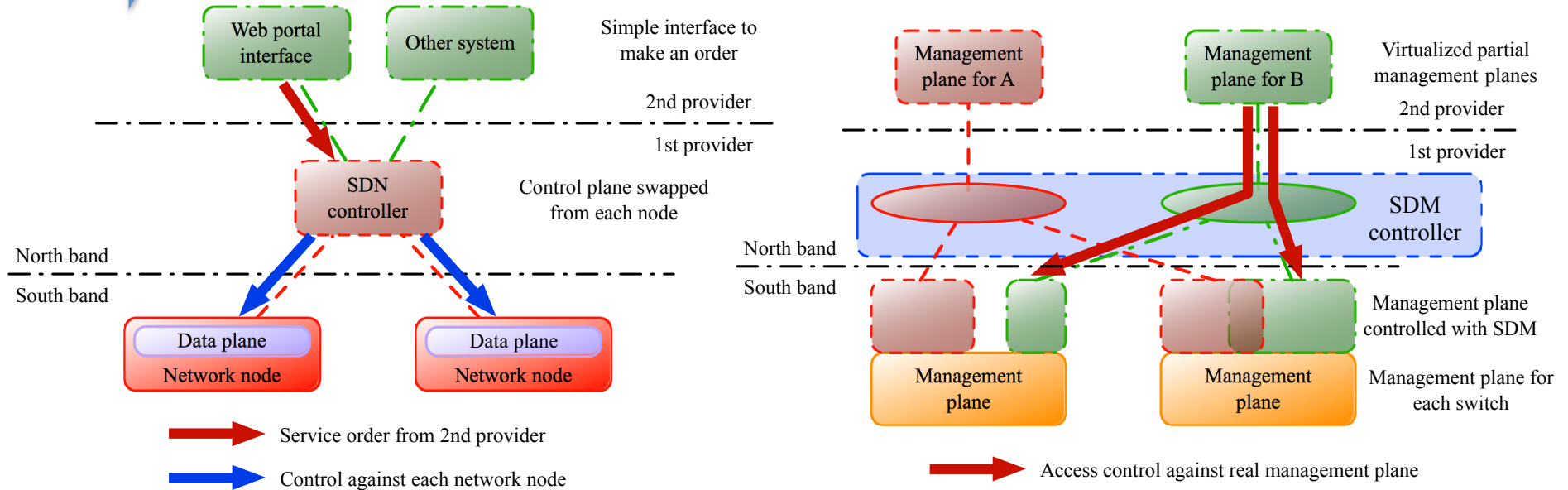
- マネジメントの競合を柔軟に解決するアーキテクチャの提案
  - Service Dependent Management (SDM)
  - マネジメントプレーンを統一ポリシーごとに分離し、  
マネジメントプレーンの一部委譲を集中的かつ柔軟に制御
- SDM のマネジメントプレーンに対するオペレーション操作時のコマンドに特化した実装の設計・実装
  - Simple Management of Access-Restriction Translator Gateway (SMART-GW)



# マネジメントプレーンの中央制御化

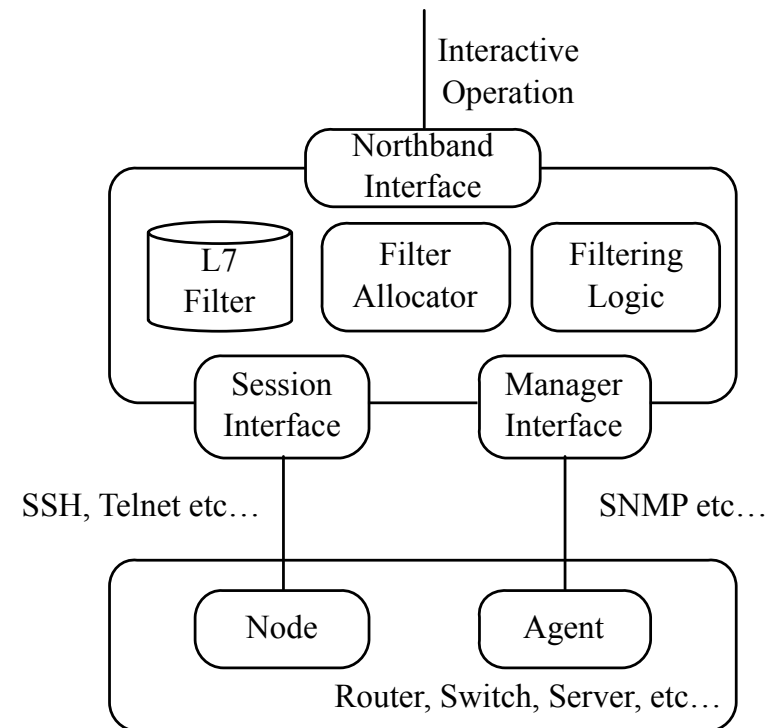
- マネジメントプレーンの中央制御化はコントロールプレーンのそれとは本質的に異なる
  - コントロールプレーン
    - プロトコルとして規定されており、インタフェースを統一可能
  - マネジメントプレーン
    - プロトコルとして厳密に規定されておらず、実装はベンダに強く依存し、インタフェースの統一化は困難

➡ L7 フィルタによるマネジメントプレーンの絞り込みにより制御



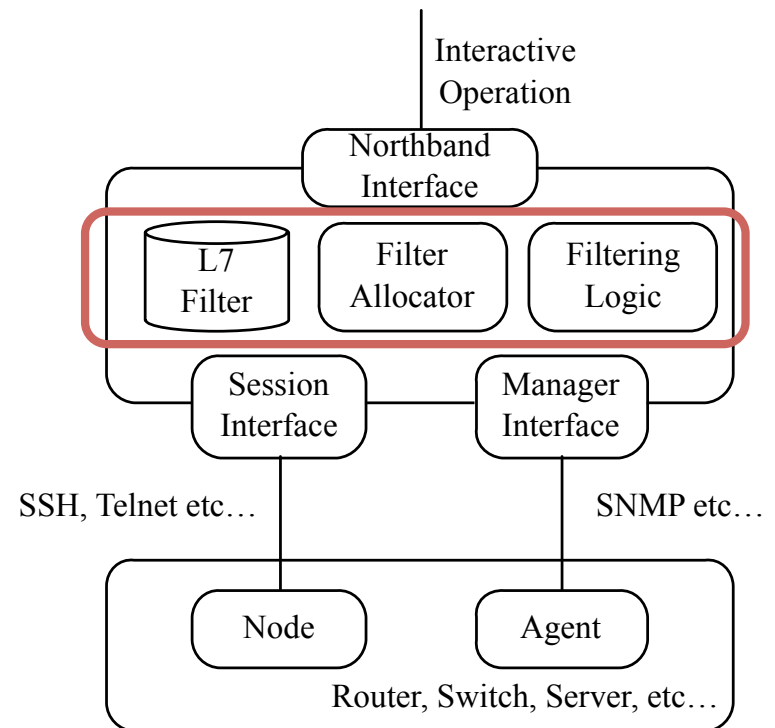
# SDM の構成要素

- North Band Interface
  - 2次プロバイダが利用する  
フロントエンドインタフェース
- L7 Filter
  - 提供するマネジメントプレーン  
の範囲を定義するフィルタ
- Filter Allocator
  - 各トラップやセッションに  
対してフィルタの割り当て方を判断
- Filtering Logic
  - Filter Allocator により割り当てられた  
フィルタの利用ロジック
- Session Interface
  - 各制御機器を操作するためのインタフェース
- Manager Interface
  - 各種トラップを受け取るためのマネージャインタフェース



# SDM の概要と構成要素

- North Band Interface
  - 2次プロバイダが利用する  
フロントエンドインタフェース
- L7 Filter
  - 提供するマネジメントプレーン  
の範囲を定義するフィルタ
- Filter Allocator
  - 各トラップやセッションに  
対してフィルタの割り当て方を判断
- Filtering Logic
  - Filter Allocator により割り当てられた  
フィルタの利用ロジック
- Session Interface
  - 各制御機器を操作するためのインタフェース
- Manager Interface
  - 各種トラップを受け取るためのマネージャインタフェース



SDM の具体的実装は **L7 Filter, Filter Allocator, Filtering Logic** を実装することで完了

# SMART-GW の概要

---

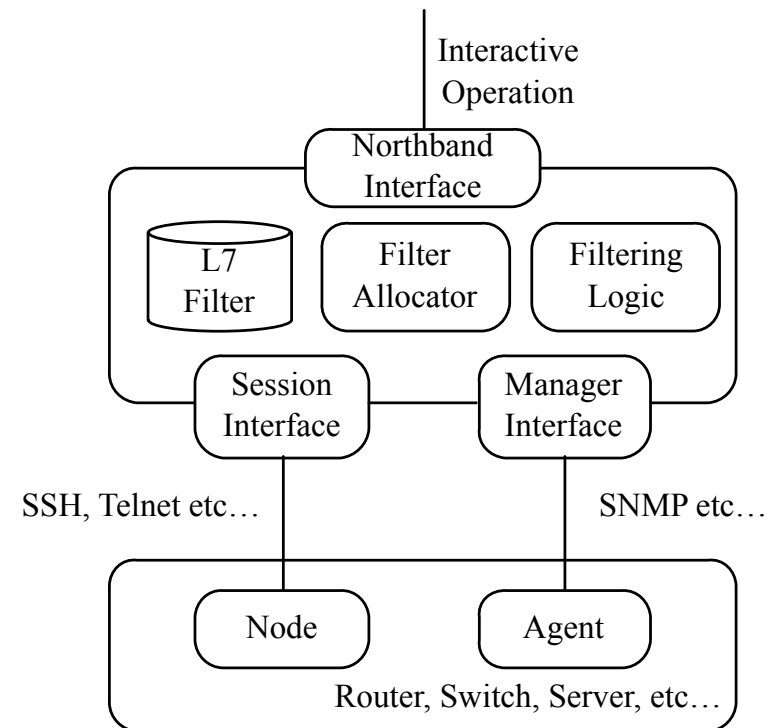
- マネジメントプレーンに対するオペレーション操作時のコマンドに特化した実装
  - コマンドの制限が可能
    - 2次プロバイダに対して提供する **マネジメント機能を制限**
  - 使用されたコマンドのログ管理機能
    - マネジメント操作記録を管理し、**Co-Management** を実現
  - 使用可能なコマンドを一元管理
    - 一括管理することにより、即時的かつ柔軟に制御可能
  - 制御対象に非依存

SDM の枠組みを用いたマネジメントを  
柔軟かつ一括に制御できる実装

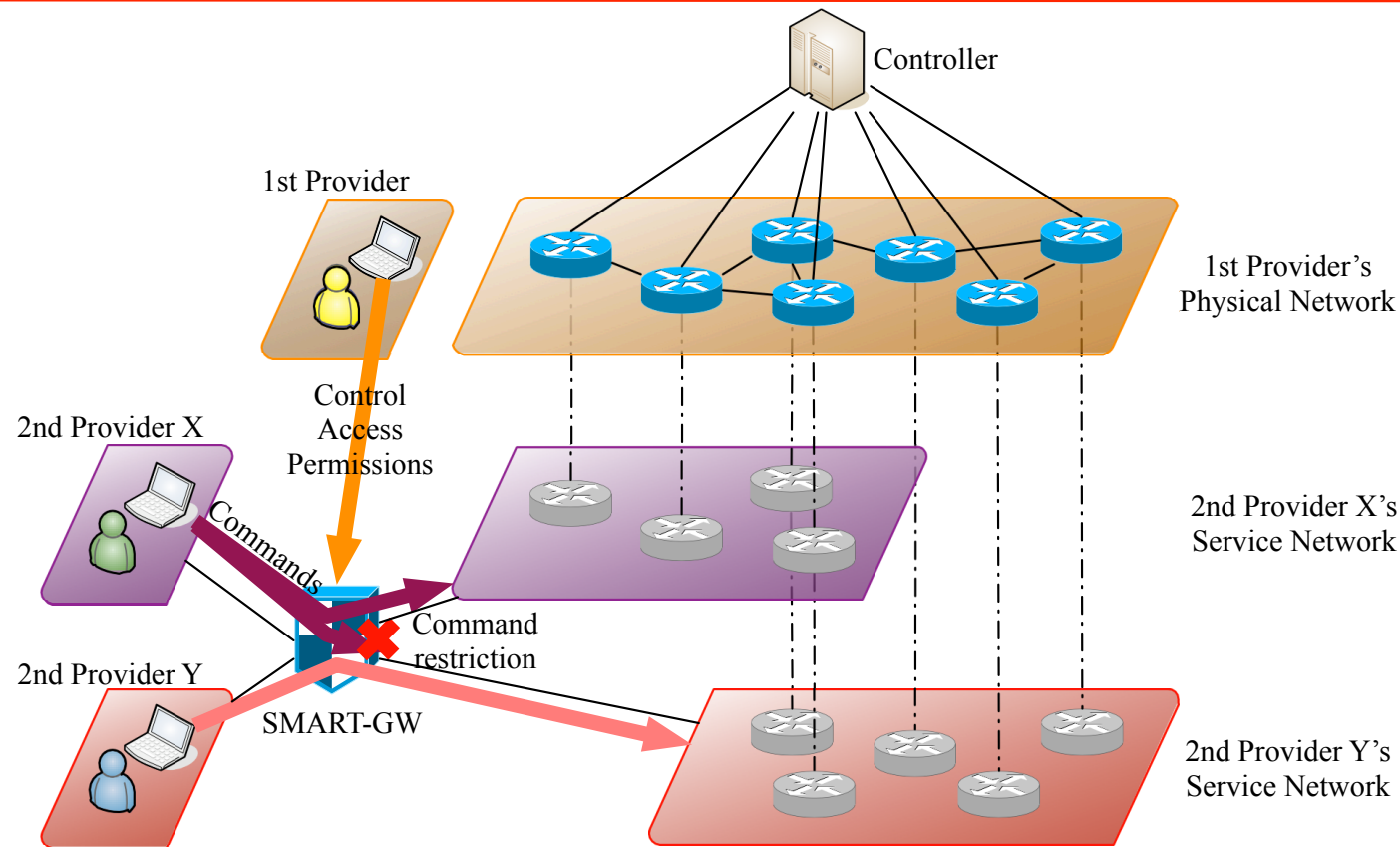


# SMART-GWの実装

- Filter Allocator
  - ユーザ名とログイン先グループに基づいてL7 Filter を適宜割り当てる
    - ユーザ名：2次プロバイダ名
    - ログイン先グループ：ノード群
- L7 Filter
  - 正規表現を用いたフィルタ
    - インタフェースの非依存性を高めるため
- Filtering Logic
  - L7 Filter に当てはまる場合のみ、コマンドを許可
  - L7 Filter が空フィルタの場合は該当するログイン先グループにアクセス権限無し

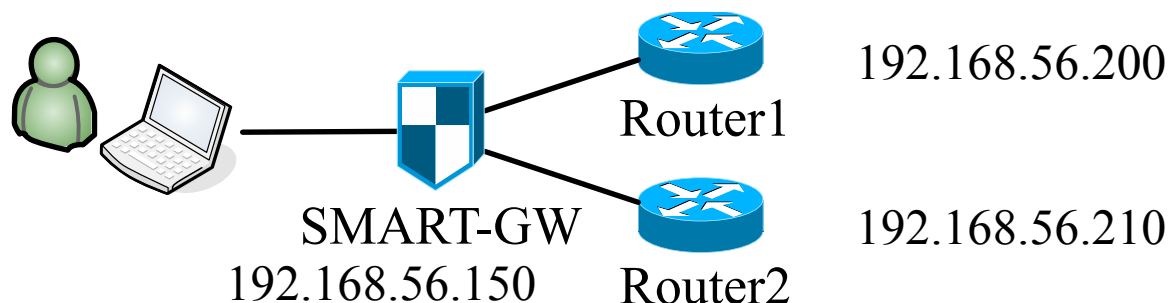


# 実システムにおける SMART-GW の導入



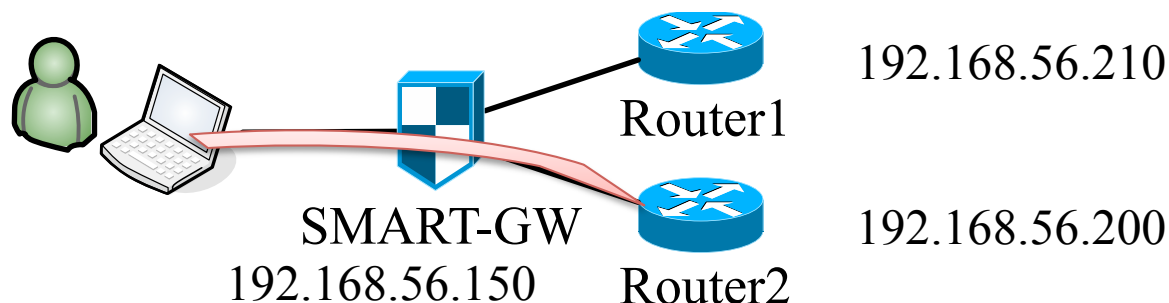
- 1次プロバイダが各2次プロバイダにそれぞれ異なる許可コマンドを一元的に管理
  - 2次プロバイダが直接インフラをマネジメント可能

# 動作確認



- SMART-GWの動作を確認
  1. SMART-GW へ test ユーザでログイン
  2. Router2 へ telnet でアクセスし、コマンド制限が働くことを確認
    - 任意の粒度で制御できることを確認
  3. Router1 に関するフィルタを追加
  4. Router1 へ telnet でアクセスし、適応したコマンド制限が働くことを確認
    - 動的かつ柔軟に制御できることを確認

# 動作確認



- SMART-GWの動作を確認
  1. SMART-GW へ test ユーザでログイン
  2. Router2 へ telnet でアクセスし、コマンド制限が働くことを確認
    - 任意の粒度で制御できることを確認
  3. Router1 に関するフィルタを追加
  4. Router1 へ telnet でアクセスし、適応したコマンド制限が働くことを確認
    - 動的かつ柔軟に制御できることを確認

## ロゲイン及びフィルタの確認

```

1. ~ (ssh)
Connection to 192.168.56.150 closed.
[nakayama@nmac ~]$ ssh test@192.168.56.150
test@192.168.56.150's password:
Last login: Thu Jan 15 10:40:56 2015 from 192.168.56.1

  _ _ _ _ _
 /  _  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
 \  _  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
/  _  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
 \  _  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \  \
  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _

Welcome to SMART GW CLI. (ver. 1.1.5)
Copyright (c) 2014 BOSCO Technologies Inc. All rights reserved.
Type 'help' or '?' for help. Use 'ctrl+c' to interrupt and clear current inputs.

SMART> admin
password:
% Authorization succeed.
SMART(admin)> exit
SMART> filter
  target group  | rule
-----+-----
****          | ^show.*$
SMART> telnet 192.168.56.200 2601

```

デフォルトルールでは show  
コマンドを許している



# Router2 におけるオペレーション

```
SMART> telnet 192.168.56.200 2601
```

telnet により Router2 へログイン

```
% Using default rule as filter of this connection.  
% Now connecting.....
```

```
Trying 192.168.56.200...  
Connected to 192.168.56.200.  
Escape character is 'off'.
```

```
User Access Verification
```

```
Password:
```

```
Router> show interface eth0
```

```
Interface eth0 is up, line protocol is up  
index 2 metric 1 mtu 1500  
flags: <UP,BROADCAST,RUNNING,MULTICAST>  
HWaddr: 08:00:27:c2:15:d9  
inet 192.168.56.200/24 broadcast 192.168.56.255  
inet6 fe80::a00:27ff:fec2:15d9/64
```

show は送信されるが、  
enable コマンドは送信されない

```
Router> enable
```

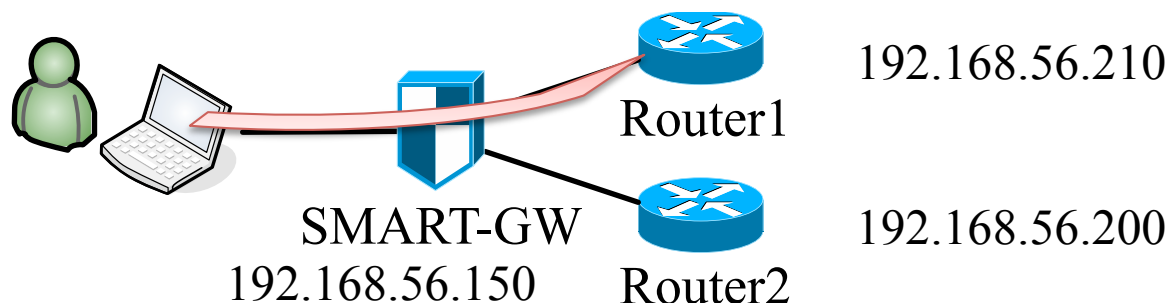
```
Command "enable" restricted because of not matching to regular expression.
```

```
Router> 
```



フィルタリング機能が正しく働いており、  
マネジメントプレーンの機能制限が実現

# 動作確認



- SMART-GWの動作を確認
  1. SMART-GW へ test ユーザでログイン
  2. Router2 へ telnet でアクセスし、コマンド制限が働くことを確認
    - 任意の粒度で制御できることを確認
  3. Router1 に関するフィルタを追加
  4. Router1 へ telnet でアクセスし、適応したコマンド制限が働くことを確認
    - 動的かつ柔軟に制御できることを確認

# フィルタの変更および確認

1次プロバイダによるフィルタ変更

```
root@narch:~  
SMART(admin)> rule add test route 1  
Input command filter rule [no input means forbid all commands].  
filter rule: ^show .* lo$  
% Succeed to add new rule.  
SMART(admin)> 
```

rule add 実行前

```
SMART> filter  
target group | rule  
-----+-----  
**** | ^show.*$
```

rule add 実行後ルールが動的に変更できている

```
SMART> filter  
target group | rule  
-----+-----  
router1 | ^show .* lo$  
**** | ^show.*$  
SMART> 
```



一括でかつ動的にマネジメントプレーンの  
委譲範囲を変更可能

# Router1におけるオペレーション

```
~ (ssh)
Trying 192.168.56.210...
Connected to 192.168.56.210.
Escape character is 'off'.
User Access Verification
Password:
Router> show interface eth0
Command "show interface eth0" restricted because of not matching to regular ex
pression.
Router> show interface lo
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
Router> 
```

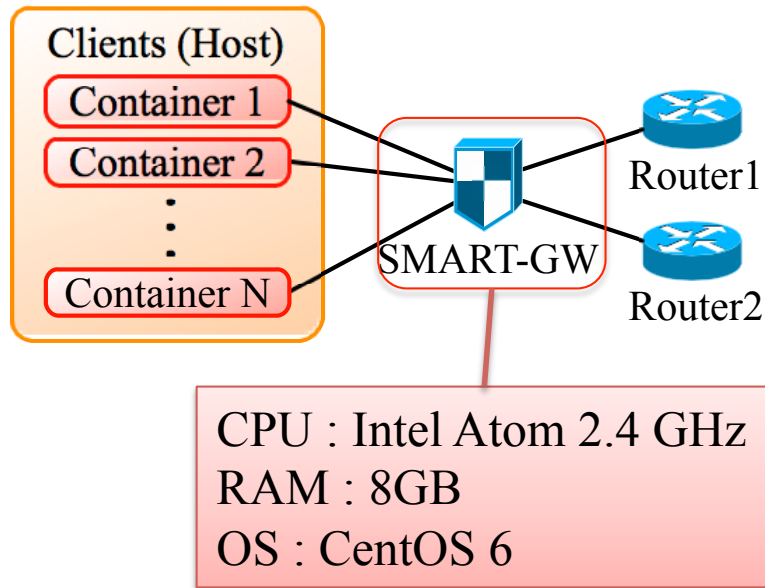
eth0 の情報は参照不可

lo の情報は参照可能



動的な設定変更が正しく反映されている

# 性能評価



Test case	Parameter		
	Clients	Commands per session	Command frequency
A	100	Avg 15 (cmd / session)	Avg 10 (cmd / min)
B	500	Avg 15 (cmd / session)	Avg 10 (cmd / min)
C	1000	Avg 15 (cmd / session)	Avg 100 (cmd / min)

† cmd : command

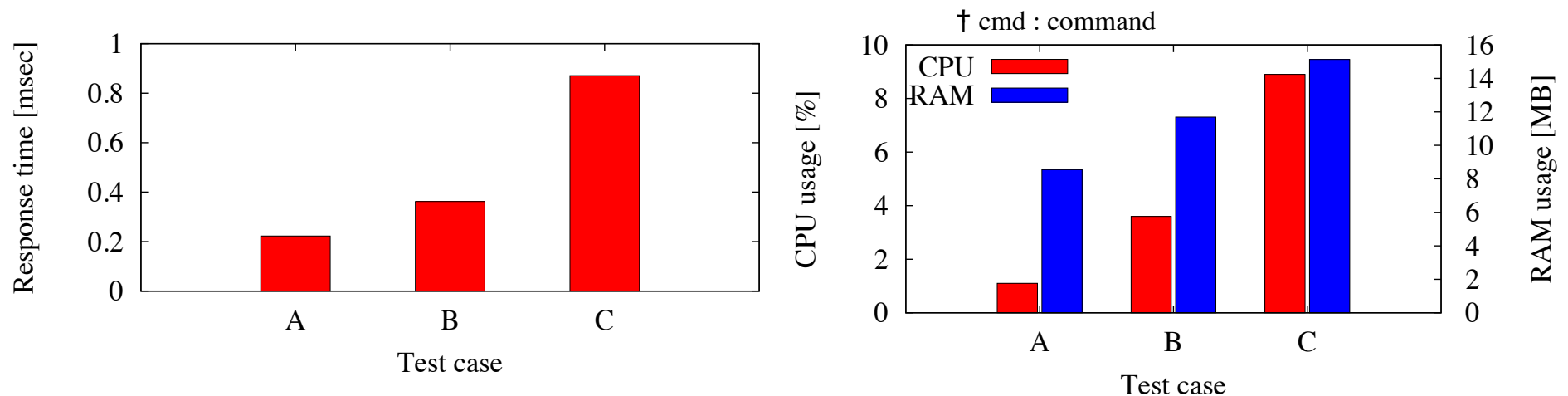
User	Filter
admin	^.*
partner	^(ping   trace   show   enable   set   help   exit).*
customer	^(show   ping   help   exit).*

- パフォーマンスの懸念点がある
    - 集中制御によるパフォーマンスの影響
    - L7-Filter に正規表現を用いることによる影響
- ➡ 実用に十分耐えうるかの性能評価



# 性能評価結果

Test case	Parameter		
	Clients	Commands per session	Command frequency
A	100	Avg 15 (cmd / session)	Avg 10 (cmd / min)
B	500	Avg 15 (cmd / session)	Avg 10 (cmd / min)
C	1000	Avg 15 (cmd / session)	Avg 100 (cmd / min)



いずれのテストケースでも  
文字入力後の応答時間は  
1msec 未満

少量の計算機資源で動作し、  
一般的な Linux マシンで  
十分運用可能



- 多量の同時セッション数にも耐える
- 多量の文字列処理にも十分耐える

# まとめと今後の課題

---

- まとめ
  - ネットワーク仮想化が及ぼすマネジメントポリシー競合問題を解決するための枠組みである SDM を提案
  - SDM のマネジメントプレーンにおけるオペレーションに特化した実装である SMART-GW を提案
    - 性能評価により実用可能性を示した
    - 実 SDN システムにて稼働中
- 今後の課題
  - SDM のフレームワーク化
  - トラップに関する具体的な実装