SNMP でネットワーク構成図を自動生成する

v1 Apr.20.2018

Atsushi Fujiwara
Rakuten, Inc.

Rakuten

自己紹介

藤原 敦史

所属： 楽天株式会社
　　社内向けプラットフォームサービスの運用チーム

担当業務：
　　以前はネットワークエンジニア
　　現在は社内システムのパフォーマンス監視など

興味があるもの：
　　データ収集、可視化、自動化
　　InfluxDB、Grafana、Fluentd

ネットワーク構成図の自動生成

inet-henge

https://github.com/codeout/inet-henge

構成管理DB を元にネットワーク構成図をブラウザ上で表示する

- 描画には D3.js を利用
- 構成管理DB はJSON形式で記述

構成管理DB からネットワーク図を自動生成する :: JANOG41

構成管理DBとは

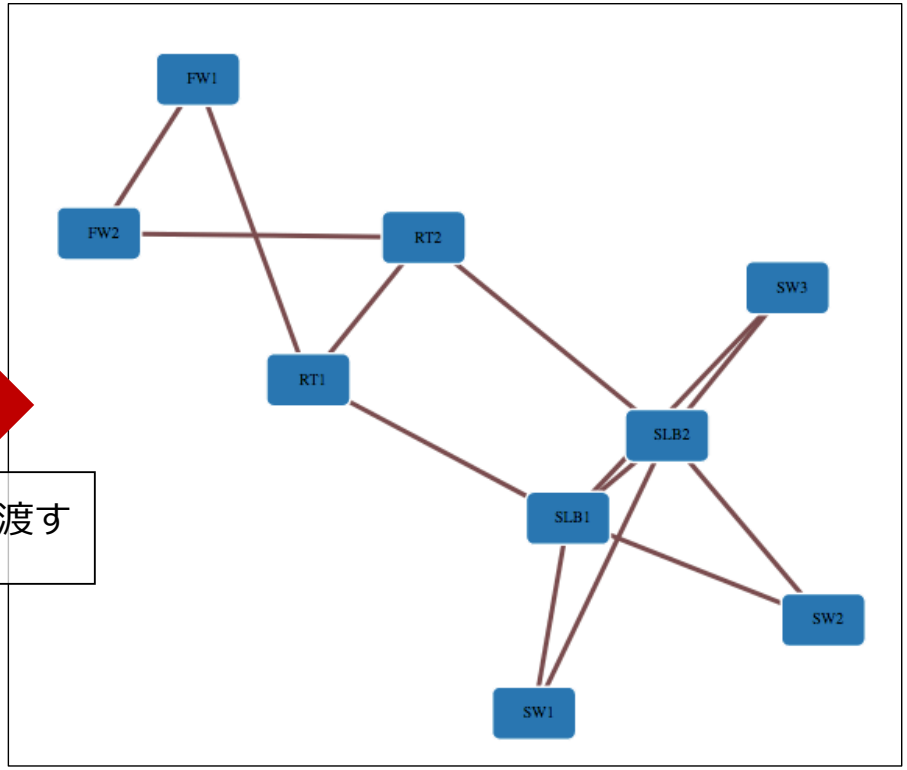
JSON でネットワーク構成情報を記述する

```
{
 "nodes": [
  {"name": "FW1"},
  {"name": "FW2"},
  {"name": "RT1"},
  {"name": "RT2"},
  {"name": "SLB1"},
  {"name": "SLB2"},
  {"name": "SW1"},
  {"name": "SW2"},
  {"name": "SW3"}
 ],
 "links": [
  {"source": "FW1", "target": "FW2"},
  {"source": "FW1", "target": "RT1"},
  {"source": "FW2", "target": "RT2"},
  {"source": "RT1", "target": "RT2"},
  {"source": "RT1", "target": "SLB1"},
  {"source": "RT2", "target": "SLB2"},
  {"source": "SLB1", "target": "SLB2"},
  {"source": "SW1", "target": "SLB1"},
  {"source": "SW1", "target": "SLB2"},
  {"source": "SW2", "target": "SLB1"},
  {"source": "SW2", "target": "SLB2"},
  {"source": "SW3", "target": "SLB1"},
  {"source": "SW3", "target": "SLB2"}
 ]
}
```

node の一覧

link の一覧

Inet-henge に渡す

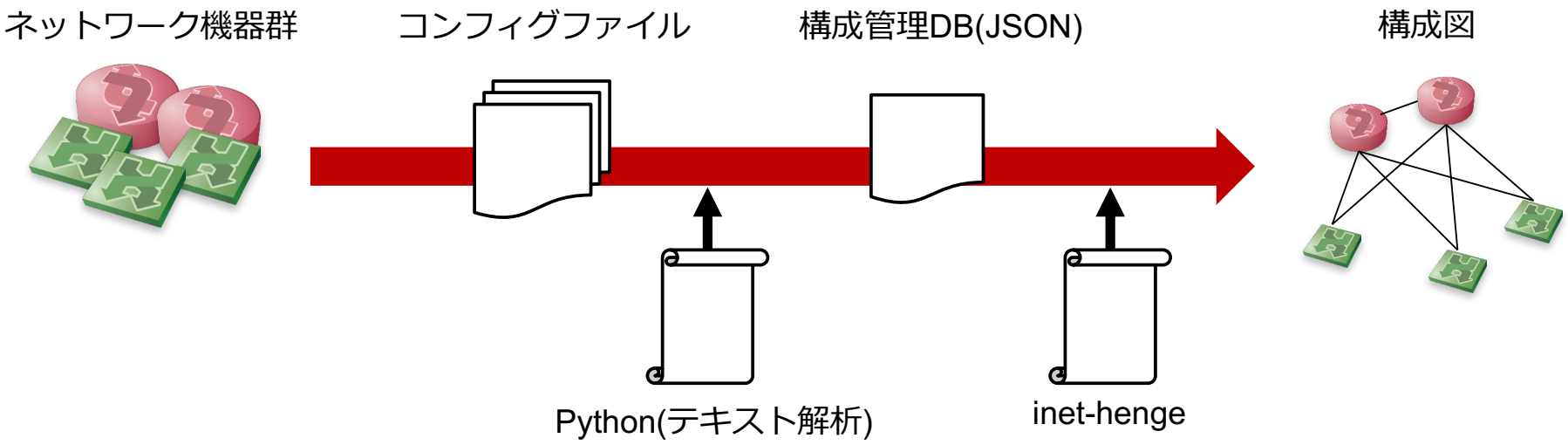


Interface Description から構成管理DBを生成する

[Interface Descriptionがプロジェクトの財産になる日 – JANOG]

ネットワーク機器の Interface Description をデータソースとする

- コンフィグファイルをプログラムで読み込んで処理
- Interface description を抽出して JSON 形式で格納

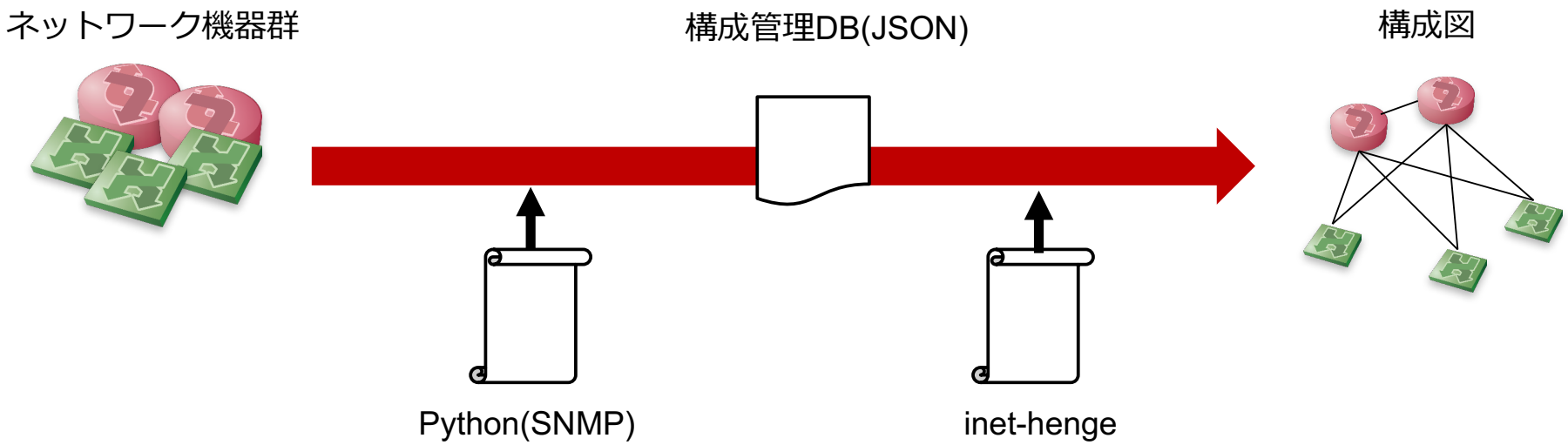


SNMPによる情報取得

SNMP でもInterface Description は取得できる

- オンラインの機器からリアルタイムに取得できる
- 標準MIB なのでベンダ/機種に依存しない

inet-archaeo: https://github.com/atfujiwara/inet-archaeo



Interface Description をどう書くか

正しい Interface Description

(ホスト名)[(区切り文字)(インタフェース名)]

こういったルールにあまり期待していない

- 先頭や末尾に”#”などをつけない
- 区切り文字は全て同じ文字を使用する
- インタフェース名は show interface などの出力と同じものを記載する

- すでに設定済みのものが大量にある
- 新規分から新しい統一ルールにする？
  => 徹底されない。。 => 繰り返されることで却って複雑になる
- それでも、ヒトが見るとなんとなくわかる

“なんとなくわかる” を実装してみる

例： “### router-1 Ethernet0/1 ###”

- ”#” やスペースはホスト名やインタフェース名に含まれない(だろう)

➡ 連続する１文字以上の“#” と “ ”で区切ってみる(/[#]+/)

正解

ホスト名： router-1
インタフェース名： Ethernet0/1

別の例

例： “router-1 (Ethernet0/1)”

- ”(” や “)” もホスト名やインタフェース名に含まれない(はず)

➡ “(” と “)” をパターンに追加(/[#()]+/)

正解

ホスト名： router-1
インタフェース名： Ethernet0/1

難しい例

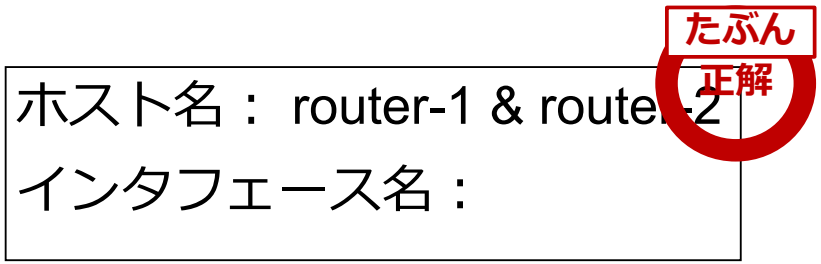
例： “### router-1 & router-2 ###”

- ”#” とスペースはホスト名やインタフェース名に含まれない(だろう)

  ➡ 連続する１文字以上の“#” と “ ”で区切ってみる(/[#]+/)

- 名前の先頭や末尾が “&” になるのは不自然

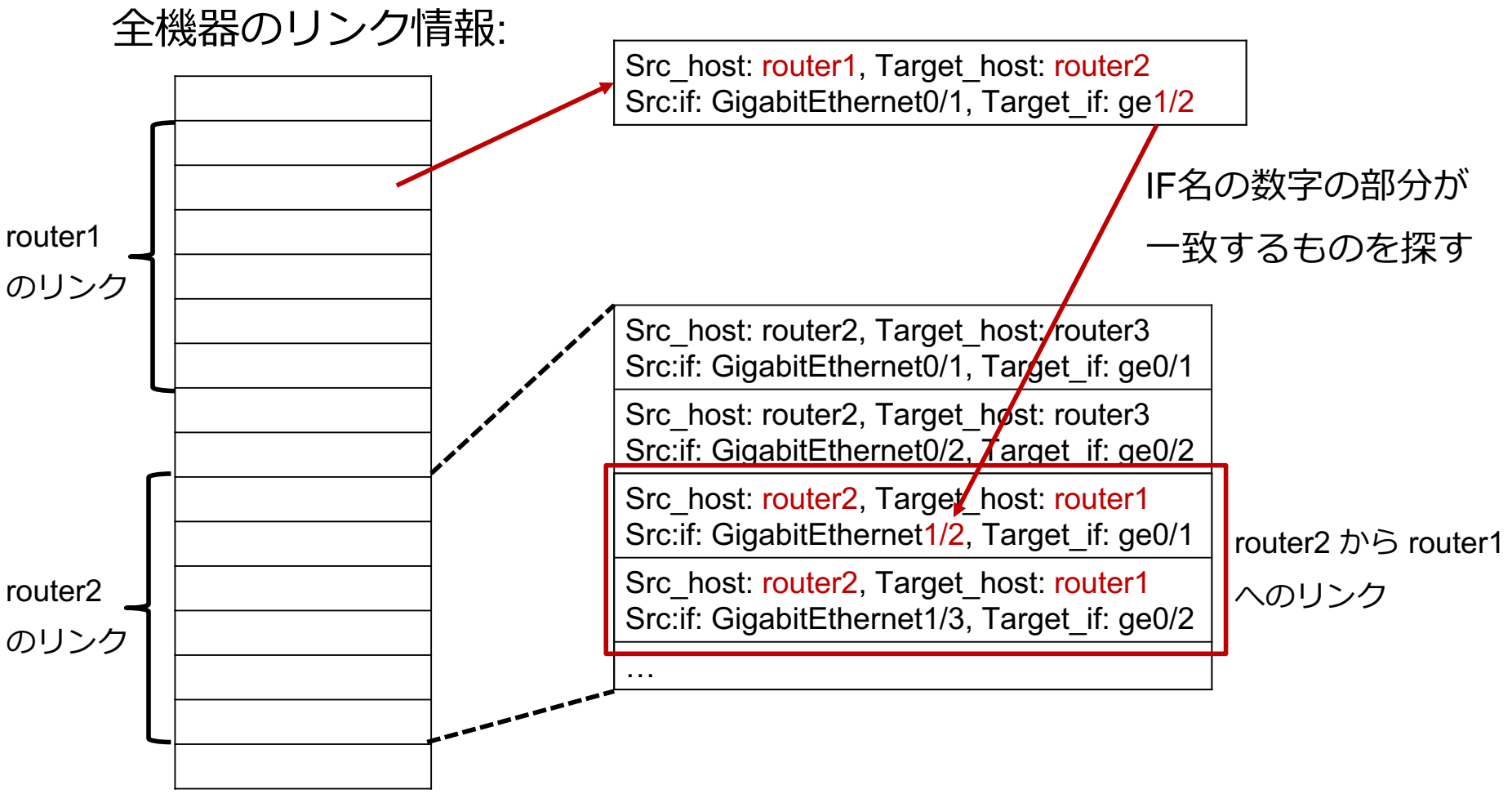
  ➡ 区切り位置が適切ではないので再度連結



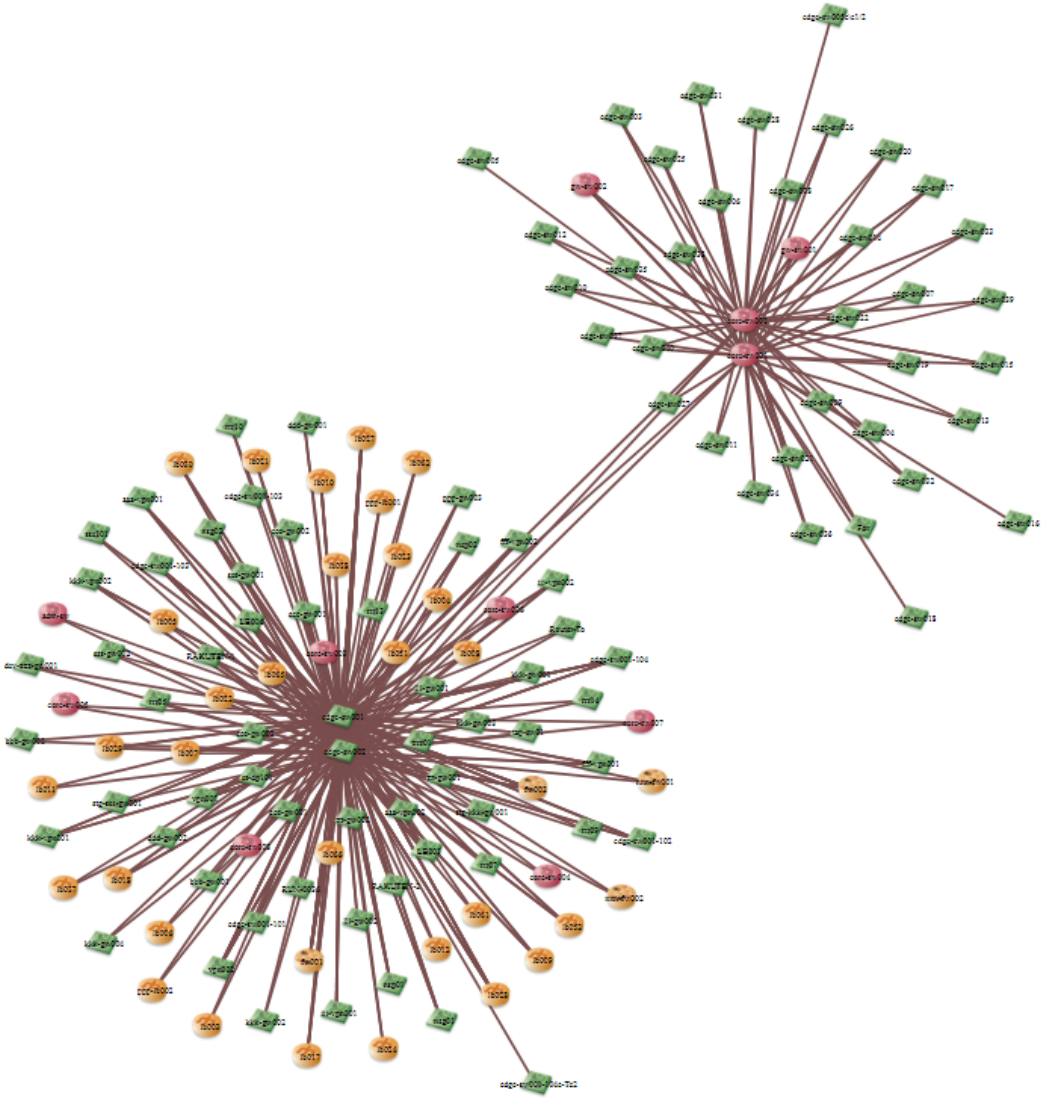
解析結果も完全には信用しない

自ホスト名: router1 自IF名: GitabitEthernet0/1	実際の値	自ホスト名: router2 自IF名: GitabitEthernet1/2
対向ホスト名: router2	おそらく正しい	対向ホスト名: router1
対向IF名: ge1/2	正しくないかも	対向IF名: ge0/1

逆方向のリンクを探して情報を補完する



自動生成した構成図



ノードもリンクも多すぎ

接続されている全ノードを取得してしまう

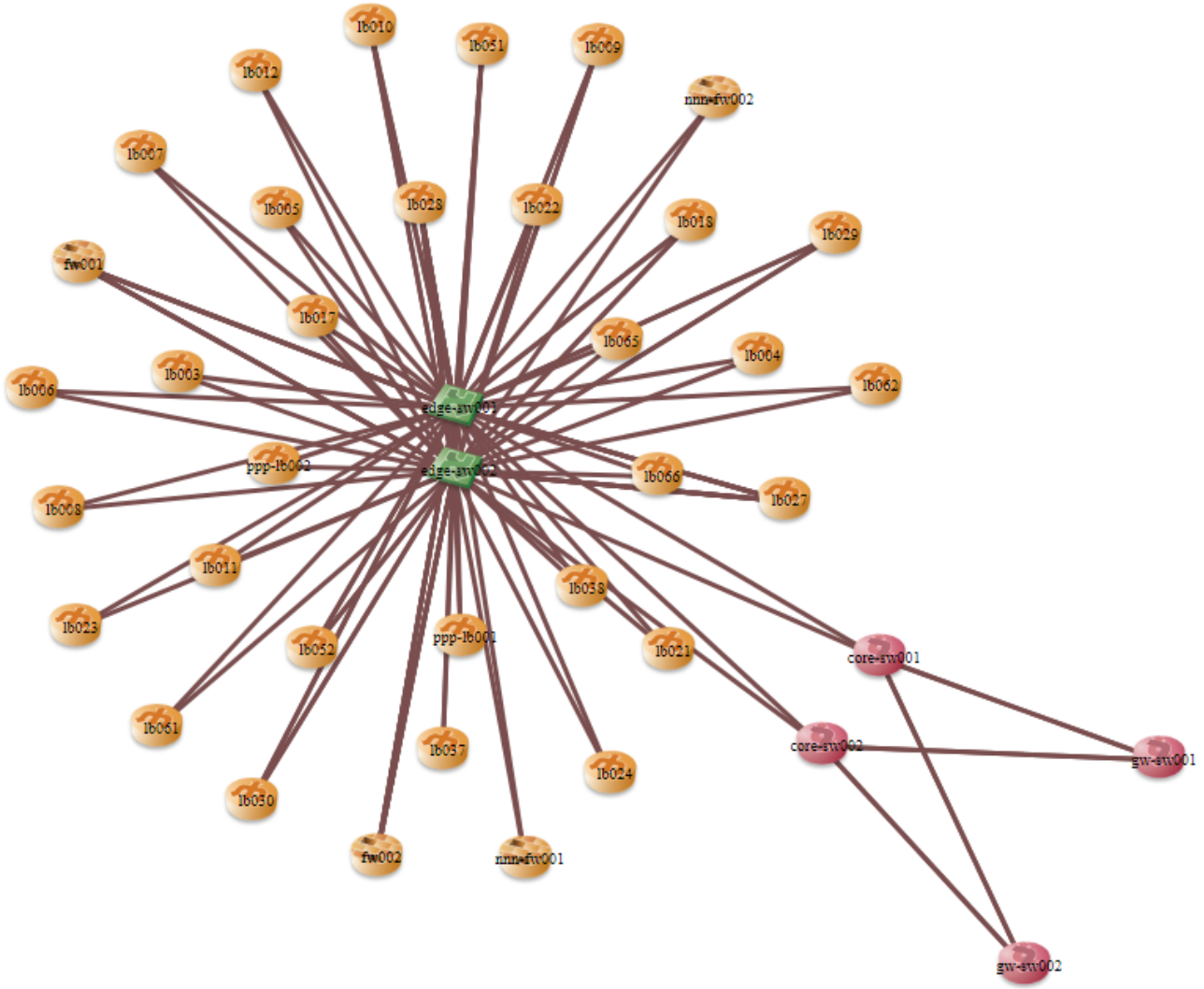
- 実際に構成図を手書きするときはスコープをあらかじめ決めている

出力に含めるノード/インタフェースを選択する

- 自ホスト名、対向ホスト名、自インタフェース名、対向インタフェース名でフィルタを記述
- 合致したものを出力に含める or 含めない

```
link_filter:
 - {src_host: ‘router[1,2]’, action: ‘include’}
 …
 - {target_host: ‘^.*$’, action: ‘exclude’}
```

自動生成し直した構成図



ノード間の複数本リンク

実際には描かれているが重なっている

回避策：

現時点ではインタフェース名の方を連結して表記するしかなさそう（未実装）

まとめ

SNMP で取得したデータを用いて inet-henge で構成図を描く

- データソースとしては標準のものなのでベンダ/機種ごとの実装は不要
- Interface Description を柔軟に解析する
- ルールベースで構成図のスコープを絞る
- Trunk 接続などを描けるようにしたい
- SNMPでデータ取得する部分を並列化すれば高速化できそう


Rakuten