FUJITSU
shaping tomorrow with you

Interface Descriptionが
プロジェクトの財産になる日
～ネットワーク構成図の自動描画～

富士通株式会社
宮崎 貴大
@JANOG41 in 広島

FUJITSU

Data-Oriented Description
（=データ指向Description）

これ↑聞いたことありますか?

自己紹介

FUJITSU

- 名前：宮崎　貴大（Miyazaki　Takahiro）

- 所属：富士通株式会社
  ネットワークサービス事業本部
  ネットワークソリューションセンター
  社会基盤ネットワークソリューション部

- 仕事
  - 対象業界：通信事業者、電力会社
  - スキルセット：PMP、ネスペ、CCIE #36811 (R&S)
  - 業務：ネットワークの提案(&獲得)～構築(&プロマネ)～保守サポート

プロローグ

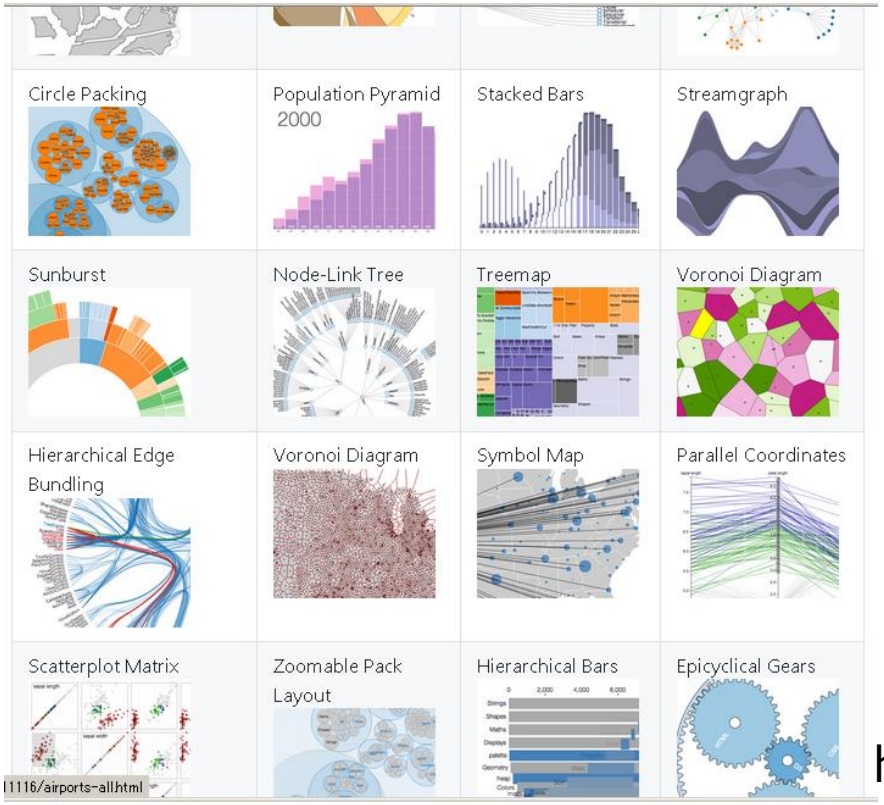
FUJITSU

- 1,500台を超える機器で構成されるネットワークがある
- 次期網の検討で、何がどう接続されているのかを把握したい
- しかし、全機器が収まる構成図は作成されていない。

ということで手近にあるもので何か作る

FUJITSU

- データサイエンティストなる人たちの間ではD3(Data-Driven Documents)がはやっている模様。
- D3.jsはJavaScriptのライブラリでデータの可視化を行うことを目的としたもので下図にあるような色々な表現形態がある。オープンソースなので無料。
- しかし、JavaScriptさわったことすらないのでどうするか・・・。



https://d3js.org/

手軽そうなツールを探してみる

FUJITSU

- D3.js系ツール“**inet-henge**” を発見
  - Githubの絵が好みなので試すことに決定
  - 詳しくは本日午後のLT（by小島さん）

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

- ↓こういう形式でデータを渡すだけ(JSON)

```
{
  "nodes": [
    { "name": “Router1" },
    { "name": “Router2" }
  ],

  "links": [
    { "source": “Router1", "target": “Router2" }
  ]
}
```

- ↑これでRouterAとRouterBとが繋がるという最小の定義。これを繰り返す。
- JavaScriptを知らなくてもネットワークトポロジ指向でD3.jsを活用できる！

データのソースをどこに求めるか

FUJITSU

■ ノードの情報とリンクの情報とがどこから手に入るだろうか？※いまある範囲で

✓ CDP/LLDP：
既にC社やJ社など、マルチベンダ構成。機能をサポートしてない装置もある。

✓ MACアドレス探索：
MACテーブル等の情報をひたすらトレースすることで何がどこに繋がっているかを
調査して元ネタをつくることはできそうだが条件とか色々と面倒そう。

✓ パラメータシート：
1台につき1シートでどのポートがどこに接続されているか等まとまっているものがある。
しかし、画像つきでにかく重い。ファイルダウンロード中に断念。∵1,000台以上あるし・・・。

✓ Interface Description：
物理IFのDescriptionはポリシー上”対向ホスト名＋ポート名”で
構成されていることを認識し、必要なものは揃っていると判断。処理もテキストなので軽い。

検討マトリックス	CDP/LLDP系 showコマンド	MACテーブル系 showコマンド	パラメータシート (.xls)	Interface Description
オフライン作業　可否	×	×	○	○
マルチベンダー対応　可否	△	○	○	○
データソース　ファイルサイズ	○	○	×	○

➢ 総合評価でInterface Descriptionに決定

Interface Descriptionおさらい

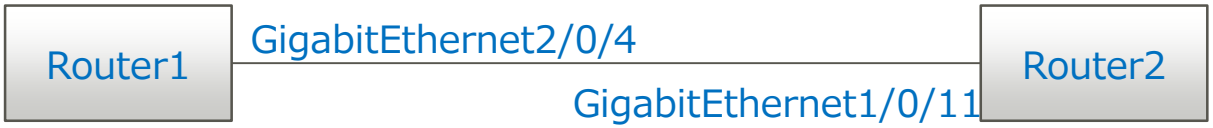
FUJITSU

- Descriptionはインターフェースごとに記述できる。

```
……
hostname Router1
……
……
interface GigabitEthernet2/0/4
 description Router2:GigabitEthernet1/0/11
 ip address 10.0.1.1 255.255.255.252
```

どのホストのどこのポートが
他のホストのどのポートに繋がっているかわかる

これが描ける



- 業界内に鉄板の様な書き方はあるか?
  - 異なる顧客でも同様のポリシーで記載されていた
  - 海外のブログをみてもホスト名＋インターフェース名でやっているのを見つけた。

- しかし、<u>なくても機器の動作に全く影響しない</u>ので、書かれなかったり、適当だったりする

Interface Descriptionのいいところ発見

FUJITSU

- 構成図あたりがメンテされなくなって放置というケースは想像に難くないが、コンフィグの中にあるDescriptionは放置されるリスクが低いように思える

- Descriptionは「なくても動く」が、見方を変えると、「情報を埋め込む」という貴重な機会を提供しているともいえる。

InterfaceDescriptionは
100円貯金くらいの意義がある気がしてくる

どうやってコンフィグから情報を抜き出して整えるか?

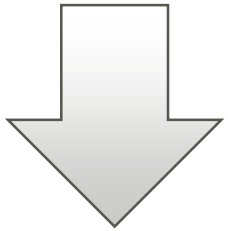
FUJITSU

- やらないとけないこと
  - フォルダにあるたくさんのコンフィグファイルを順次読み込む
  - インターフェースのdescriptionを順次チェックしてJSON形式に格納していく
  - 上記の繰り返し処理をするプログラミング言語を選ぶ
- 文字列を抜き出すくらいならどの言語でもできそう。
  ただ、正規表現が使える言語がいい。
- 最近、Pythonがハヤりなのでこれにしておく。

取り組み始めて3時間後

FUJITSU

- 50行程度のPythonコード(by 初心者)で当初の目標を達成できた。
  （他ベンダー対応するときもこの程度の追加で足りそう）



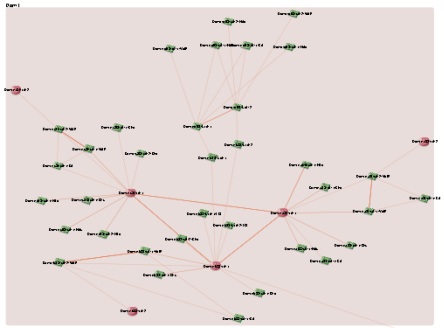
近所のちょっとプログラミングできる人に御願したら
きっと引き受けてくれるはず！

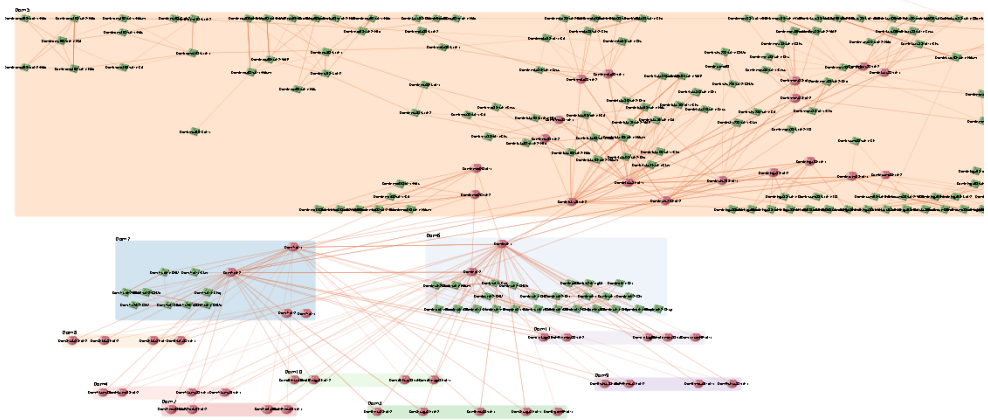
「初心者が50行程度書けばいいレベルだからよろしく」といって捕まえましょう。

Output （できたもの）

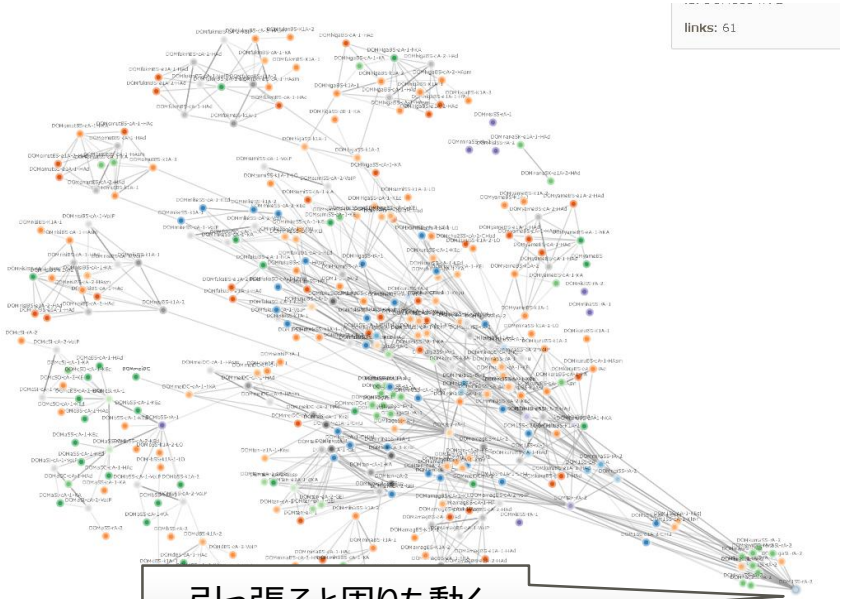
- 200台くらいのノード数では下図のような状況。ドラッグ＆ドロップで移動できる。
  ※発表用に新規に環境を作ってコンフィグ収集しました・・・
- 拡大すると仕込んでおいたポート名(Ethernet xxx)が浮かび上がる
- ホスト名でアイコンを選択してみた　※コンフィグの中身みて選択するのも◎
  緑＝L2スイッチ赤＝ルータ。
- 同じデータで他のOSSも試してみた。https://github.com/netjson/netjsongraph.js/tree/master

inet-henge.js





netjsongraph.js



引っ張ると周りも動く
その後ゴムのように戻る

参考：前のスライドのトポロジ描画時のリソース消費

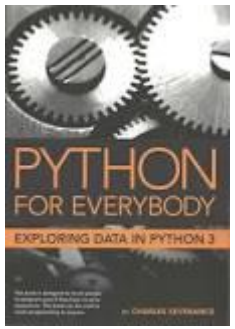
FUJITSU

- 機種：LIFEBOOK A746/N（ノートPC）
- OS：Windows 10 Pro （64bit）
- CPU：インテル® Core™ i7-6600U プロセッサー 2.60GHz [2コア/4スレッド]
- グラフィックアクセラレータ：Intel® HD Graphics 520（CPU内蔵）
- メモリ：32GB
- ブラウザ：Firefox
- ノード数=324 、リンク数=1,400 (コンフィグ⇒JSON形式への加工は10秒程度)
- inet-henge.js
  - 描画にかかる時間：2分半
  - 最大負荷：CPU 27%(平均15%くらい)＆ GPU 10% 程度
  - 消費メモリ：200MB 程度（アイコン等に依存すると思う）
  - 描画後にノードを動かすときの負荷：CPU15% ＆ GPU5% 程度
- netjsongpraph.js
  - 描画にかかる時間：20秒 程度
  - 最大負荷： CPU 30% & GPU 20% 程度
  - 消費メモリ：100MB程度
  - 描画後にノードを動かすときの負荷：CPU25% ＆ GPU20% 程度

参考：オープンな学習教材『みんなのパイソン』

FUJITSU

- “***Python for Everybody:Exploring Data in Python 3***”
  Open Educationの一環で無償の教育コンテンツ



- 様々な形態があり、電子版は基本的に0円。
  https://www.py4e.com/
- 文学作品から特定の文字を抜き出すとかしながら少しずつすすんでいく。
- 入門だけどしっかり教えてくれる。英語は平易で読みやすく、ノリがいい。
- “Extracting data using regular expression”のあたりを読めば
  今回のような目標はだいたい達成できる。あとはWeb検索でOK。

プログラミングして思ったDescription記載時の注意点

FUJITSU

■清く正しいDescriptionがプログラミングを簡単にする。

- “ホスト名”　＋”区切り文字”＋ ”ポート名” が物理IFの鉄板だと思う。
  Hostname ＋ Delimiter ＋ Port-No.
  ※サブとか、VLANとかでいろいろあるけど物理はこれでいいかな。

- 大事なのは
  - ホスト名とポート名との間に区切り文字（delimiter）があること。
  - その区切り文字が他で使われていないこと。
    （区切り文字を乱用しないこと）
  - Interface Descriptionもコンフィグ作成時しっかりレビューすること。

◎ Programming-friendlyなDescriptionの例

FUJITSU

- この↓例では区切り文字=":" 。簡単&綺麗に抜き出せる。

```
・・・・・・・・・
hostname Dom1-Router1
・・・・・・・・・
・・・・・・・・・
interface GigabitEthernet2/0/4
 description Dom1-Router2:GigabitEthernet1/0/11
 ip address 10.0.1.1 255.255.255.252
```

descriptionがある行の処理の流れ
① スペースで区切る。["description"と"Dom1～:1/0/11"に分かれる]
② ①で取り出した要素の後半(Dom1～)を" : "で区切る
③ ②の処理の前半がホスト名で、後半がIF情報なので適切な変数にセット。

- 区切り文字、":"(コロン)あたりがいいのではないかと思いますが皆さんどうですか？

× プログラマから逃げられるDescriptionの例

FUJITSU

- この↓例では区切り文字=”-” 。

```
・・・・・・・・・
hostname Dom1-Router1
・・・・・・・・・
・・・・・・・・・
interface GigabitEthernet2/0/4
 description ##Dom1-Router2-GigabitEthernet1/0/11##
 ip address 10.0.1.1 255.255.255.252
```

- △ 要素取り出すときに“##”がわりと邪魔。“#”はプロンプトにもよく使われている。
- × ホスト名のなかで”-”が使われていて、区切り文字としても“-”がある。最悪。

結局、Data-Oriented Description って何?

FUJITSU

Data-Oriented Description（データ指向Description）とは・・・

- ✓Descriptionを活用可能なデータとして認識し、
- ✓プロジェクトにとって有益で
- ✓Programming-friendlyなDescriptionを
- ✓いま/未来のプロジェクトメンバーのために残していく

という考え。機器のホスト名も広義のDescriptionに含まれる。
※Web検索で引っかかるものとは別物なので注意。

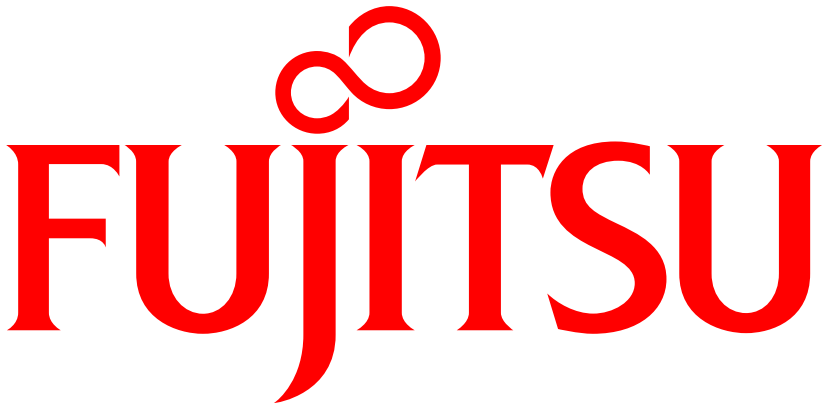
まとめ と 伝えたいこと

FUJITSU

- Description活用の一例として巨大なネットワークトポロジの可視化ができる。
- 一貫した記述ルールは視覚効果を与える事ができる。
  例）「ホスト名に”RT”を含むとルータ」⇒可視化時にルータのアイコン付与
- 区切り文字を大事にするとProgramming-friendlyなDescriptionに！

Data-Oriented Description

業界全体で蓄積していきましょう!



shaping tomorrow with you