

ネットワークテスト自動化スクリプティングの 昔と今

2024年7月4日

キーサイト・テクノロジー株式会社

松田 健

自己紹介

- 名前: 松田 健
- 所属: キーサイト・テクノロジー株式会社
ソリューションエンジニアリング本部

- 業務内容:
皆さんが“IXIA”と呼んでいる
ネットワークテスターなどを担当する
セールスエンジニアです

過去 (2001~2010頃?) には Sler の中のネットワーク
エンジニアとしてネットワーク試験に携わる機会も
多くありましたが、あまり自動化できていませんでした



IxNetwork

- ネットワークインフラの
総合パフォーマンステストソリューション
- 高負荷ステートレストラフィック生成
- ルーティング・SDN など多彩な
コントロールプレーン疑似



IxLoad

- データ・ビデオ・音声などの
アプリケーション通信を QoE を測定
- 高スケーラの HTTP, TLS, IPsec

100M から 800G までのハードウェアプラットフォームと
パブリッククラウド対応の仮想アプライアンス



BreakingPoint

- 多彩なアプリケーショントラフィックと
攻撃を疑似
- NGFW・IPSなどのセキュリティ機器や
DPI の機能と性能を検証



CyPerf

- クラウド・分散型ネットワークに対応した
ソフトウェアソリューション
- ゼロトラスト・SASE・TLS-VPN・コンテナ

内容

- 自動化スクリプティングのトレンドの移り変わりとはトラフィックジェネレーター
- Open Traffic Generator (OTG) プロジェクト

なぜテスト自動化

テストを自動化するモチベーション

- 手動での反復的なタスクを減らし、テストの速度と効率を向上させる
- 同じシナリオを何度も正確に再現する
- 人的リソースの節約, 人為的なミスの混入を防ぐなど

ネットワーク機器ベンダー

製品開発の中で昔からテスト自動化を必要とし、実践していた

Quality



Quality

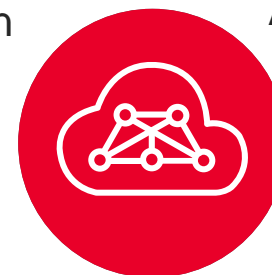
Conformance



ネットワークオペレーター

市場のニーズに応じて新しいサービス・機能を迅速に展開するためにテスト自動化の必要性が高まってきた

Network Design



Availability

Interoperability

DevOps/NetOps



SLA



目的・観点・テストケース・実施頻度
共通する部分と異なる部分

自動化スクリプティングのトレンド

テストケース自動化の進化



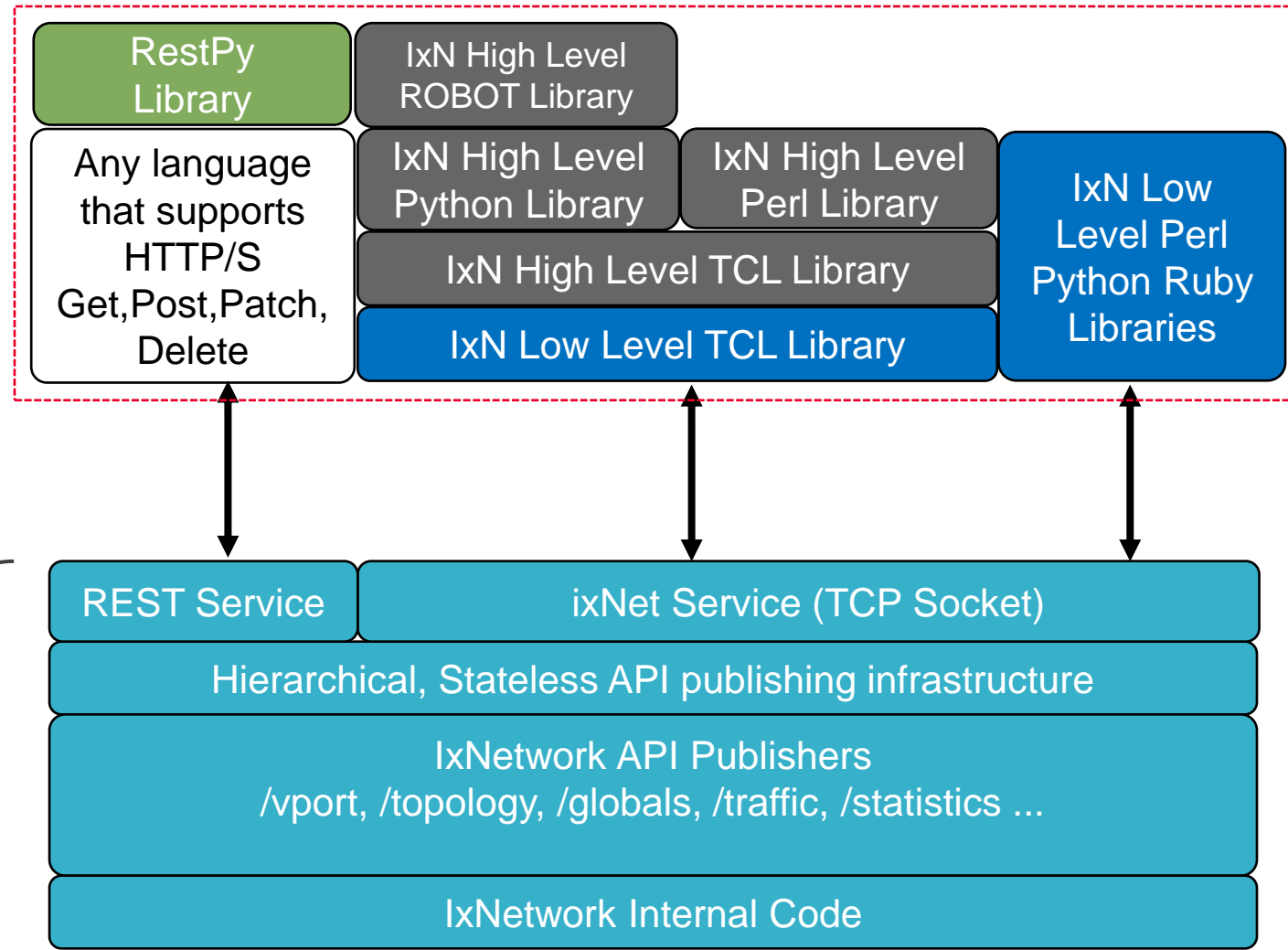
Python, REST API, JSON, YAML, Declarative, テストフレームワーク, CI/CD, ...

‘テスト’の自動化にとどまらず
より広範な自動化の中でいろいろなものと連携できる
オープンな仕組みと技術が求められるようになった

トラフィックジェネレーターは自動化のためのAPIを(複数)備えている

- ネットワーク機器ベンダーやオペレーターのテスト自動化ニーズに応えるため、さまざまなAPIを追加してきた
- 古くからのユーザーは膨大なテストスクリプトのライブラリを持っているため、古いAPIを使い続ける例も多い

IxNetwork の内部



現状はPythonが主流?

- シンプルで読みやすい言語
- コミュニティとリソースの充実 → 情報の入手しやすさ
- 広範なライブラリとツールをサポート → いろいろな処理を簡単に実装可能
- 強力なテストフレームワークやテストツール → pytest, Robot Framework, pyATS など

(AI の進化で言語の違いは問題ではなくなるかも...)

(参考) 最近のトラフィックジェネレーター自動化

IxNetwork RestPyの場合

```
from ixnetwork_restpy import SessionAssistant

# create a test tool session
session_assistant = SessionAssistant(IPAddress='127.0.0.1',
    LogLevel=SessionAssistant.LOGLEVEL_INFO,
    ClearConfig=True)
ixnetwork = session_assistant.Ixnetwork

# create tx and rx port resources
port_map = session_assistant.PortMapAssistant()
port_map.Map('10.36.74.26', 2, 13, Name='Tx')
port_map.Map('10.36.74.26', 2, 14, Name='Rx')

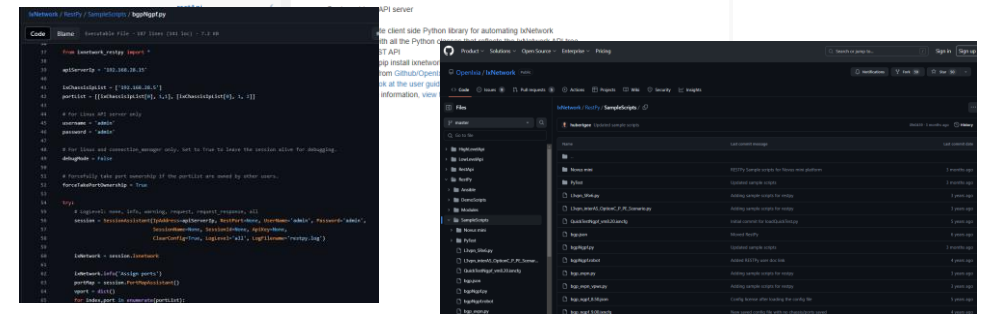
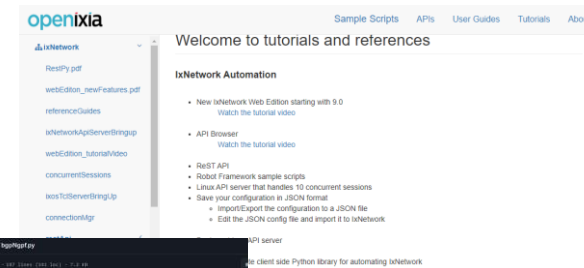
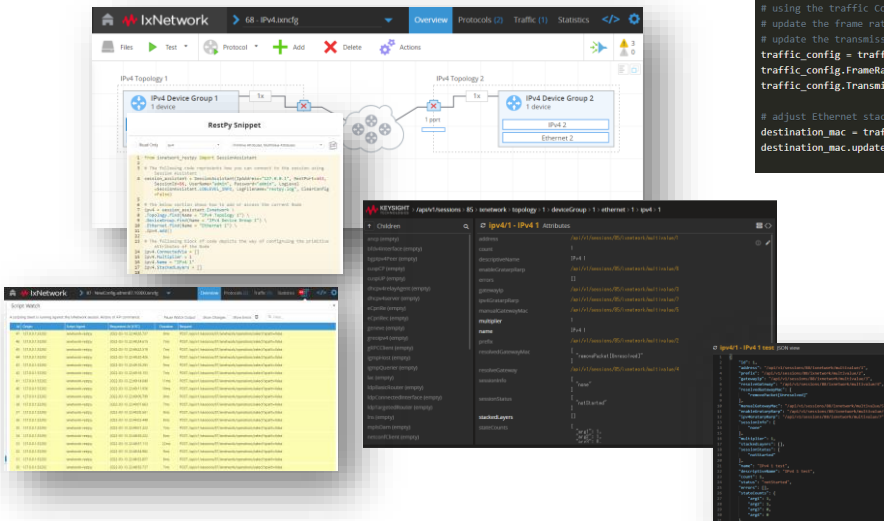
# create a TrafficItem resource
# TrafficItem acts as a high level container for ConfigElement resources
# ConfigElement is a high level container for Individual HighLevelStream resources
traffic_item = ixnetwork.Traffic.TrafficItem.add(Name='Traffic Test', TrafficType='raw')
traffic_item.EndpointSet.add(
    Sources=ixnetwork.Vport.find(Name='Tx').Protocols.find(),
    Destinations=ixnetwork.Vport.find(Name='Rx').Protocols.find())

# using the traffic configuration resource
# update the frame rate
# update the transmission control
traffic_config = traffic_item.ConfigElement.find()
traffic_config.FrameRate.update(Type='percentLineRate', Rate='100')
traffic_config.TransmissionControl.update(Type='continuous')

# adjust Ethernet stack fields
destination_mac = traffic_config.Stack.find(StackTypeId='ethernet').Field.find(FieldTypeId='ethernet')
destination_mac.update(ValueType='valueList', ValueList=['00:00:fa:ce:fa:ce', '00:00:de:ad:be:ef'])
```

IxNetwork-RestPy (2020年～ 公式サポート)

- PyPI に登録 (pip コマンドで導入可能)
- REST API の Python ラッパー
- オブジェクト指向
- API ライブラリとドキュメントを完備



Web UI, API Browser, RestPy Snippet など
スクリプト開発・デバッグを支援するツール群

RestPy を使った自動化に関するチュートリアルと
サンプルスクリプトを公開するサイト OpenIxia

Open Traffic Generator (OTG)とは

<https://otg.dev/>

トラフィックジェネレーターの
データモデルと Northbound API を定義するプロジェクト

- オープン
- インテントベース
- ベンダー中立
- 宣言的

OTG対応ツール

Ixia-c Community Edition

- コンテナベースのトラフィックジェネレーター
※ IxNetwork のコンテナ版ではない

Keysight Elastic Network Generator (KENG)

- Ixia-c の商用版 + Ixia HW を OTG で制御

Magna

- OpenConfigプロジェクトによるオープンソースのOTG 実装

TRex

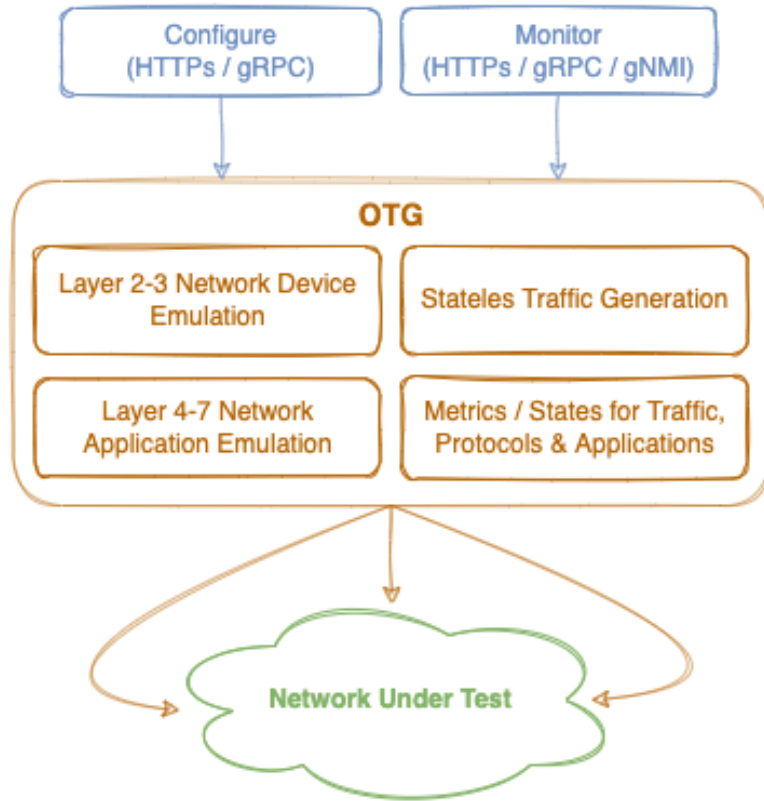
- Cisco Systemsによるオープンソースのトラフィックジェネレーター
(snappi-trexを経由して OTG で TRex を制御可能)

OTG を活用しているオープンソースプロジェクト

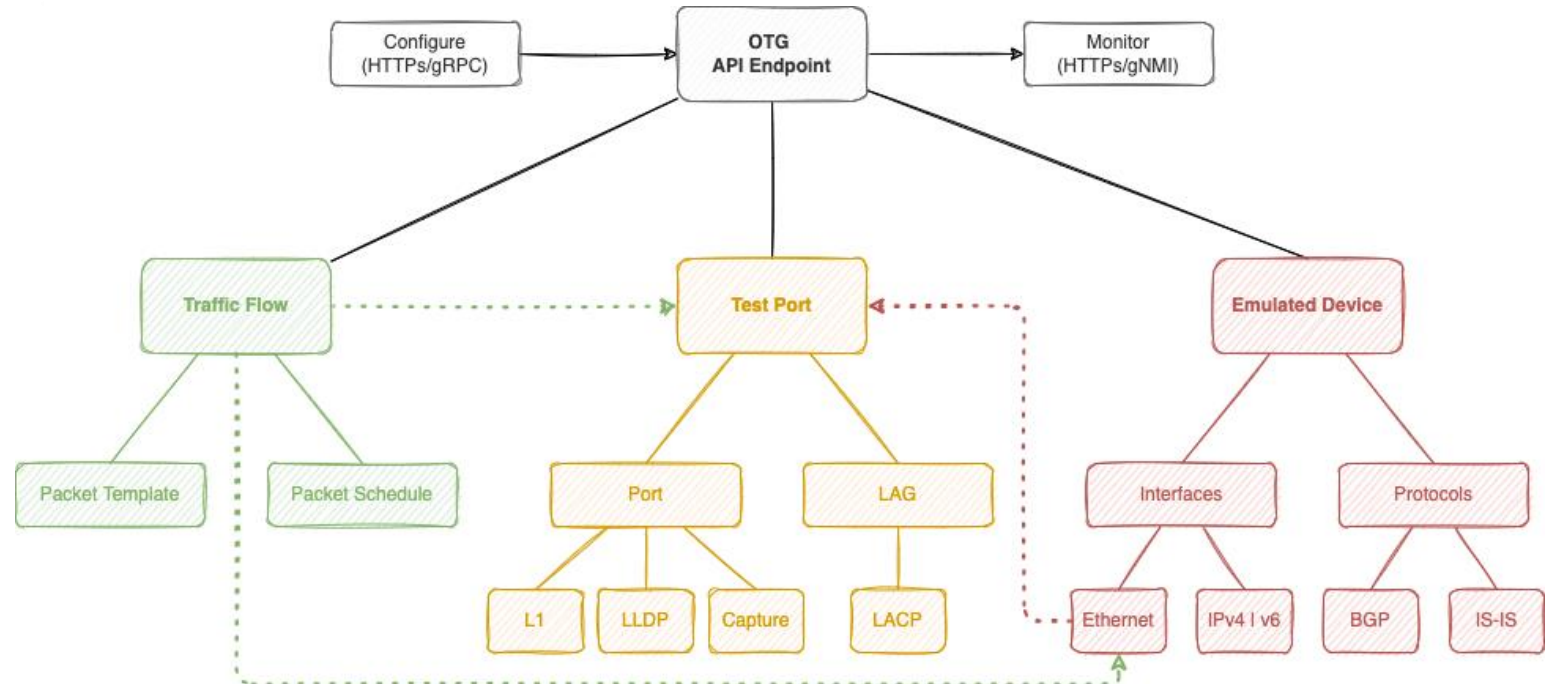
- [OpenConfig Feature Profiles](#)
- [SONiC Testbed](#)
- [SONiC-DASH CI Pipeline](#)

OTG API と OTG データモデル

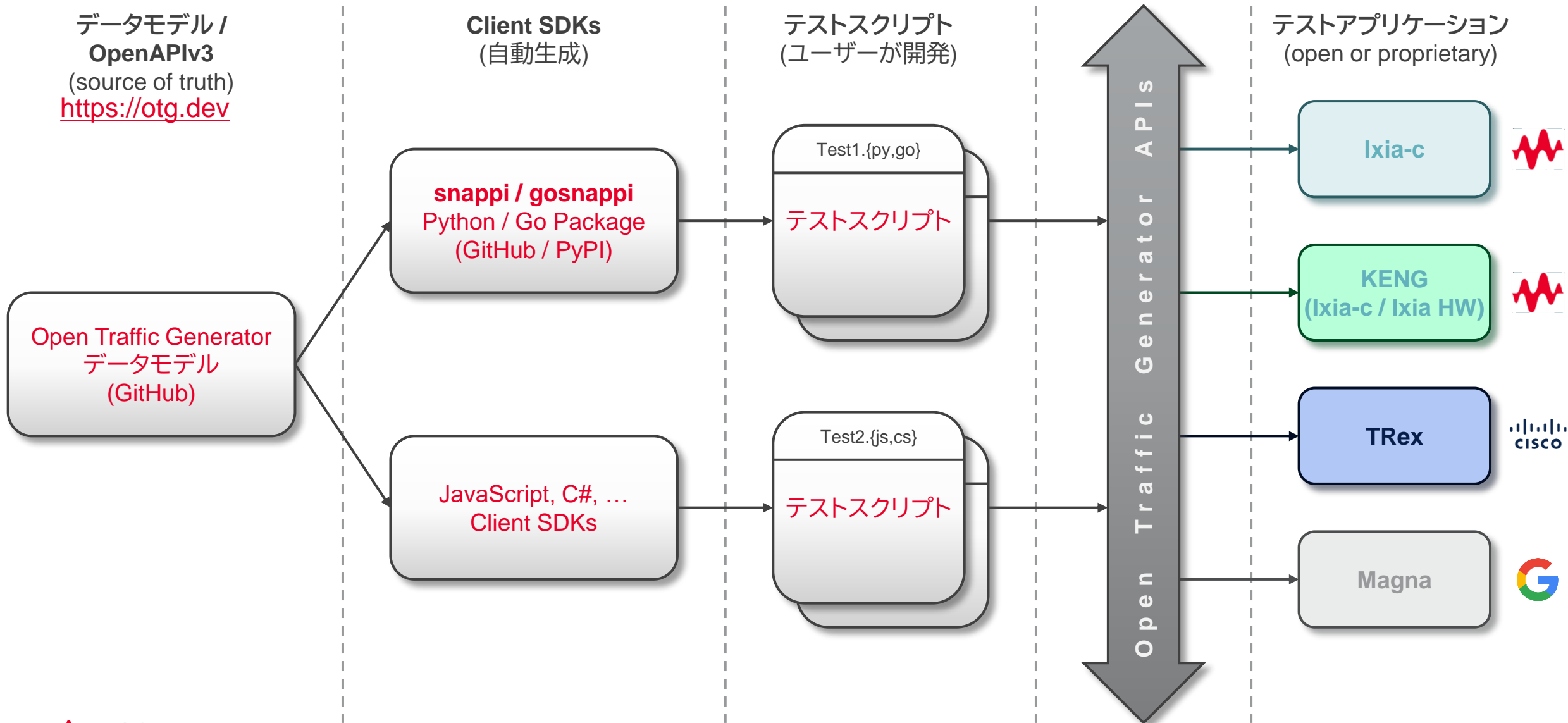
OTG API Surface



OTG L2-3 Model Hierarchy



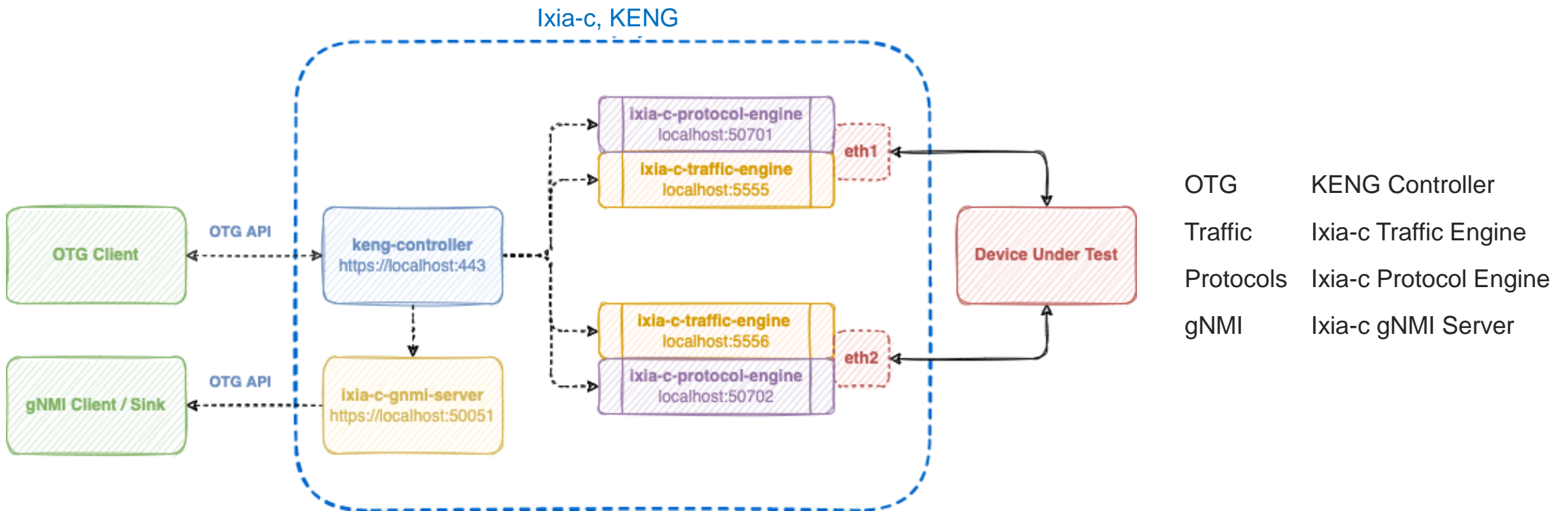
Open Traffic Generator API



OTGに対応したトラフィックジェネレーターの例

ixia-c: コンテナ版L2-3トラフィックジェネレーター

<https://ixia-c.dev/>

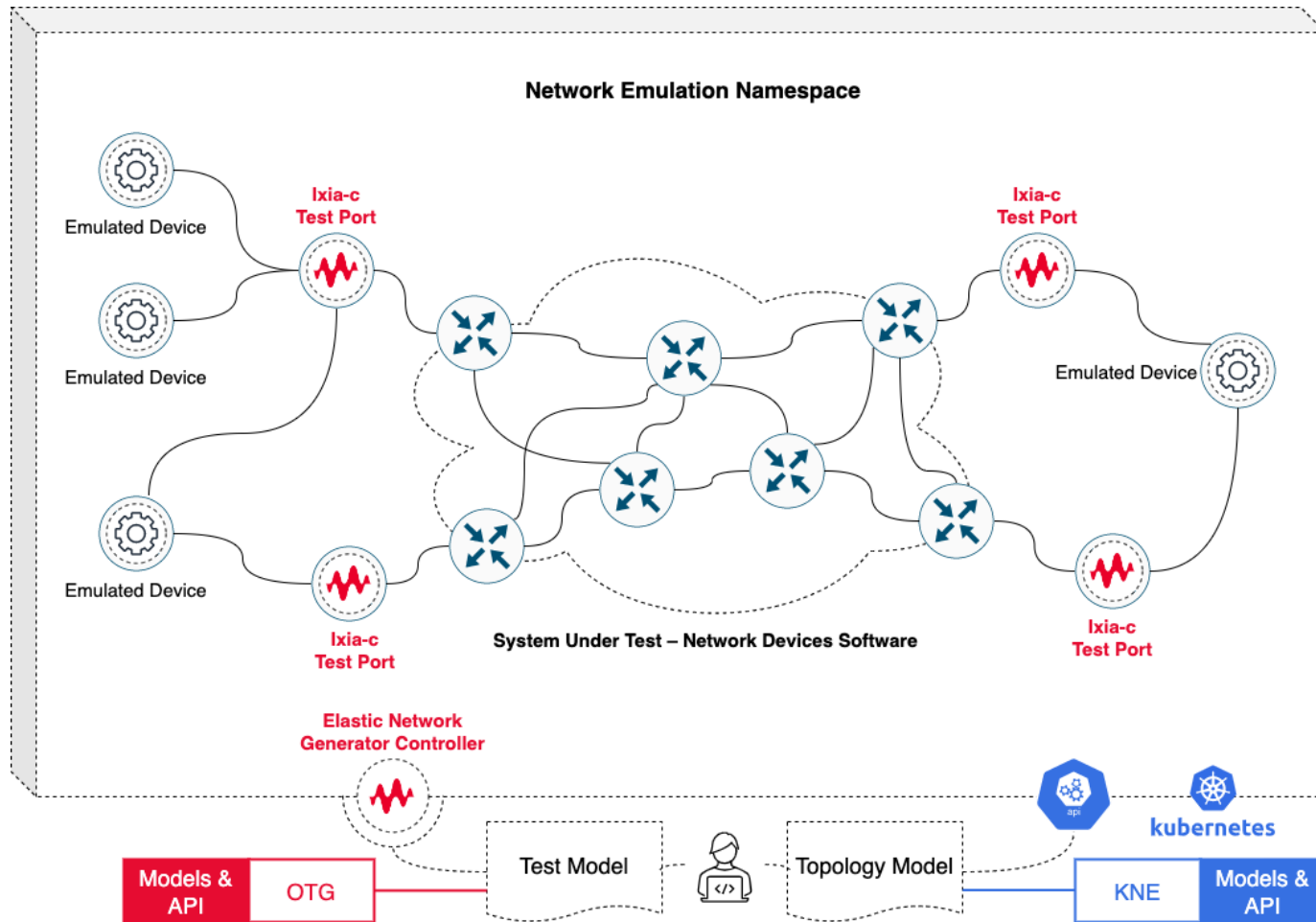


(GUI も CLI も持たない, OTG専用)

Ixia-c Community Edition* は
ユーザー登録やコスト無しで利用可能

(* コントロールプレーンプロトコルはARP/NDP/BGPのみ,
スケール/パフォーマンス制限あり)

ネットワークエミュレーション環境でテスト実行



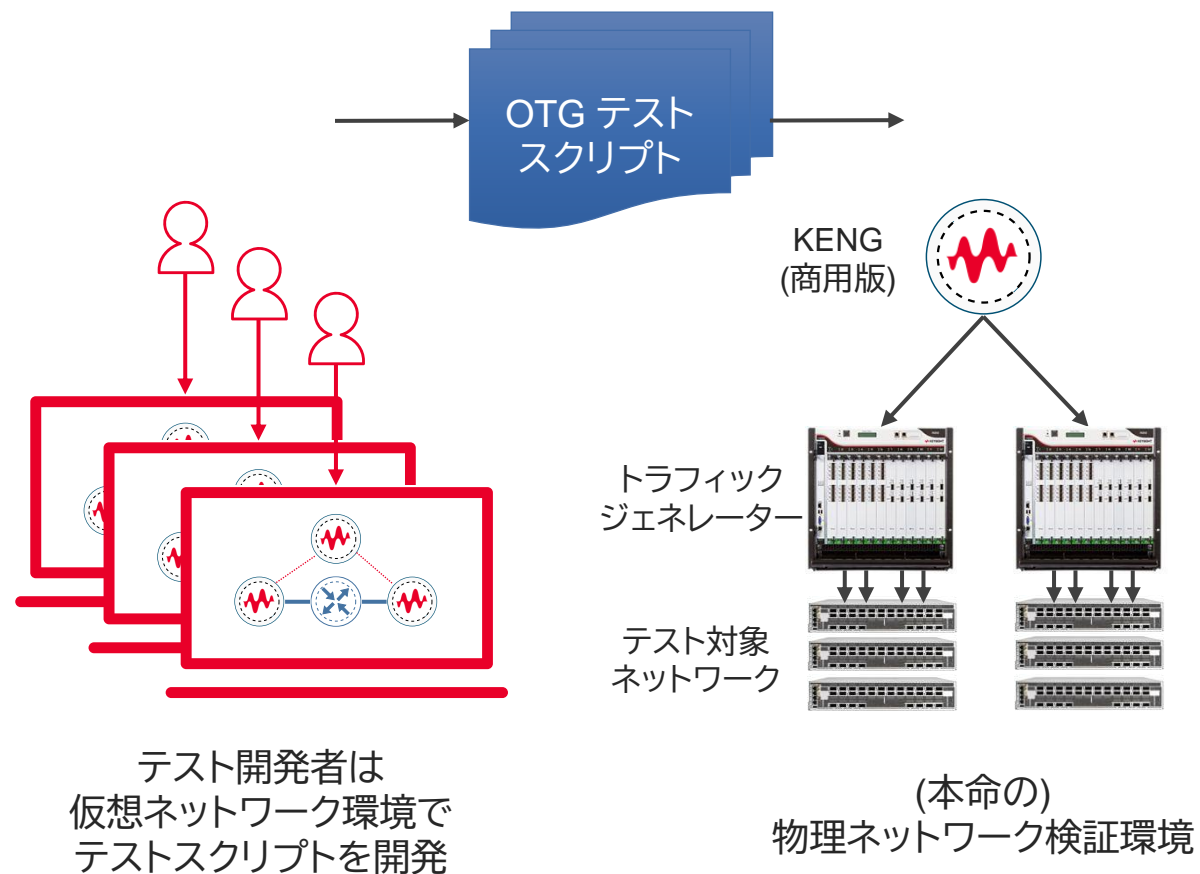
トポロジー・オーケストレーション
(仮想のネットワーク検証環境)

- K8s – KNE (Kubernetes Network Emulation)
- Docker – Containerlab, netlab
- VMs – CML, GNS3, EVE-NG, netlab

OTG 対応のトラフィックジェネレーターを
仮想ネットワーク検証環境の中に

- Ixia-c, KENG

仮想エミュレーション環境で開発したテストを物理ネットワーク環境で実施



HW無しでテストスクリプトを開発可能

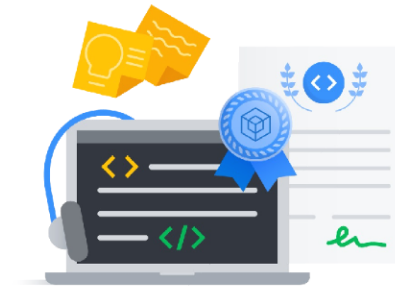
ただし

- テスト実行ツールやテストベッドオーケストレーションツールの共通化
- DUTのSW版とHW版の両方の入手

も必要

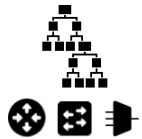
CI pipeline: OpenConfig Test Suite

Why? Cross-company compliance.

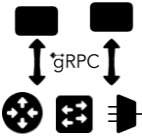


- Moving compliance away from human interpretation to code.
- Reproduction of scenarios in a packaged way.
- Ability to plug in different vendors.

NANOG™



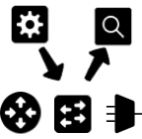
Common Data Models



Management Protocols



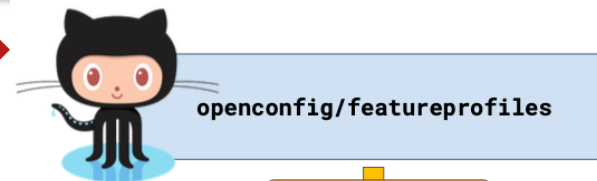
Streaming Telemetry



Testing and Compliance



“Compliance as a Code”
Test Suite



GitHub Actions

ixia-c

DUT

KNE

VM

Emulating Network Topologies in k8s
https://storage.googleapis.com/site-media-prod/meetings/NANOG86/4573/20221017_Shakir_Emulating_Network_Topologies_v1.pdf
<https://youtu.be/mAThruRU7hA>

OTGが広まると...

- 仮想ネットワークエミュレーションとソフトウェアトラフィックジェネレーターをより活用できる
 - ✓複数のテスト環境を並行運用 (開発者1人につき1つ)
 - ✓物理ネットワーク環境の準備が整うのを待たずにテストを開発
- テストシナリオやテストスクリプトの公開・共有が進む
 - ✓オープンソースプロジェクトのテストワークフローやテストスクリプトを再利用できる
- テスト実行ツールやテストフレームワークについてサードパーティーが参入し易くなる
 - ✓トラフィックジェネレーター別の実装が不要になる
- ネットワークCIの実現に近づく

お聞きしたいこと

- テスト自動化のために使用している(使用したい)テストフレームワークや言語は?
- ネットワーク検証環境として仮想のネットワークエミュレーション環境を使っていますか?
 - 検証を自動化していますか?
 - その中にトラフィックジェネレーターも組み込まれていますか?
- テスト自動化の中にネットワークテスター(トラフィックジェネレーター)を組込むにあたって障害となっていることはありますか?

