

ホームネットワークにおける HTIPに基づくネットワークトポロジ検出システム

藤巻 伶緒^{1,a)} リム 勇仁^{1,b)} 丹 康雄^{1,c)}

概要: 現在, ホームネットワークのトポロジが複雑化する一方で, 運用管理を目的とした HN 接続構成特定プロトコル (以下, HTIP) が策定されている. しかし, HTIP を用いてトポロジの情報を収集しても, トポロジを検出することは自明でない. 従来, SNMP で収集した情報を用いたトポロジ検出アルゴリズムが提案されている. 本稿では既存のトポロジ検出アルゴリズムが HTIP を利用しているホームネットワークにおいて適用できるか検討を行い, トポロジを検出し適切な形式で記述するホームネットワークトポロジ検出システムについて述べる.

キーワード: ホームネットワーク, HTIP, トポロジ検出, ネットワーク記述モデル

Network topology discovery system based on HTIP in home network

REO FUJIMAKI^{1,a)} YUTO LIM^{1,b)} YASUO TAN^{1,c)}

Abstract: Recent years, the topology of home network is getting complicated. On the other hand, an Home-network Topology Identifying Protocol (HTIP) has been formulated for operation management. However, even if topology information is collected using HTIP, it is not obvious to detect the topology. Previously, a topology discovery algorithm using information collected by SNMP has been proposed. In this paper, we study whether the existing topology discovery algorithm can be applied to home network using HTIP. We describe the home network topology discovery system.

Keywords: Home Network, HTIP, Topology Discovery, Network Description Model

1. はじめに

近年, 情報家電やモバイル端末などが一般家庭に普及し, 異なるネットワークや様々な機器が混在することにより, ホームネットワークのトポロジが複雑化 (図 1) するような状況となってきた. 例えば, 端末は有線接続されていても, ホームルータとの間に無線や電力線通信区間が入る場合も珍しくない. ネットワークに関する知識のない一般ユーザにとって, このような複雑なネットワークから自身で障害発生箇所を切り分け, 不具合を解決することは困難

である. 従って, ホームネットワークの運用管理を円滑にするため, トポロジ情報を自動的に検出し, ユーザや運用管理者に適切な形式で提供することが必要とされている. そのため, ホームネットワークに接続されたエンド端末やネットワーク機器 (以下, NW 機器) から, トポロジを検出するための断片的な情報を収集する仕組みが実装されている. その仕組みの一つに, HTIP (Home-network Topology Identifying Protocol) がある. HTIP はホームネットワークのトポロジを検出するためのプロトコルであり, 一つのリンクレイヤブロードキャストドメインについてのみ有効である. しかし, この HTIP で収集してきたトポロジ情報の扱い方についての学術的な議論はほとんどされておらず, トポロジを検出する手法は規定されていない. 他にも, SNMP (Simple Network Management Protocol) や

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

a) fujimaki@jaist.ac.jp

b) ylim@jaist.ac.jp

c) ytan@jaist.ac.jp

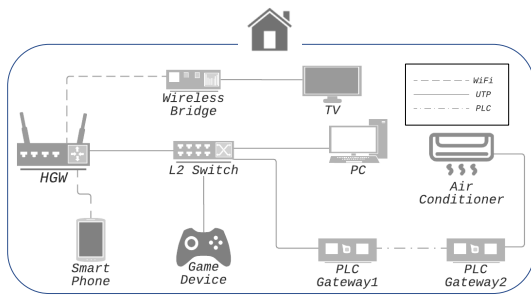


図 1 ホームネットワークのトポロジ例

Fig. 1 An example of user's home network configuration

LLTD(Link Layer Topology Discovery)などのプロトコルを利用して、ネットワークのトポロジを検出する方法もあるが、何れの方法も複雑なトポロジを形成し、多種多様なデバイスが接続されるホームネットワークの現状に即していない。

また、HTIPを用いて収集し検出されたトポロジ情報を活用するためには、適切な形式でトポロジ情報を記述する必要がある。従来、大規模なネットワークの構築や運用を容易化・自動化するために、ネットワーク記述モデルによるトポロジ情報の記述が行われてきた。しかし、ホームネットワークのトポロジ情報を形式的に記述して取扱うような必然性はまだ高くないとみなされており、実例はほとんど存在しない。更に、企業ネットワークやデータセンターのネットワークと比較して、使われている技術の多様性、過去からのシステムの拡張を重ねたことによる見通しの悪さ、全体を把握している設計者や管理者の不在といったホームネットワーク特有の問題もあり、データセンター用の技術を直接転用できない可能性も高い。

これらの問題に対して、本稿では、ホームネットワークに接続された各機器より収集されたトポロジ情報からトポロジを自動的に検出し、ネットワーク記述モデルによって検出したトポロジ情報を記述する手法をトポロジ検出システムとして提案する。2章では、ホームネットワークに接続された各機器からの情報の収集について示す。3章では、収集された情報を用いてトポロジを検出するアルゴリズムについて述べる。4章では、検出されたトポロジ情報を記述するためのネットワーク記述モデルについて述べる。5章では、トポロジを自動的に検出した後、トポロジ情報を形式的に記述し保持するトポロジ検出システムについて述べる。6章ではまとめと今後の課題について述べる。

2. トポロジ情報の収集

本章では、HTIPが実装されたホームネットワークに接続された各機器から収集する機器情報及び接続構成情報の収集について述べる。

2.1 HTIP

HTIPは、ホームネットワークのトポロジを検出するための情報を収集するプロトコルである[1]。また、リンクレイヤブロードキャストドメインにおいてのみ有効である。HTIPにおいて、接続されるエンド端末にはUPnP Device ArchitectureのControlled Deviceが実装(L3Agent)、またはLLDP Agent(Transmit only)が実装(L2Agent)され、ネットワーク機器(NW機器)にはLLDP Agent(L2Agent)が実装されている必要がある。また、Managerが各Agentから機器情報及び接続構成情報を収集することでホームネットワークのトポロジを検出する。Managerは、ホームネットワーク内の任意の端末に存在することを想定している。機器情報は、各Agent毎に管理されており、少なくとも以下の四つの情報から構成される。その他に、障害発生箇所の切り分けに有用な情報として、チャンネル使用状態情報、電波強度情報、ステータス情報を通知することが可能である。

- 区分
 - 機器の種別を表し、例えば”TV”や”PC”等の種別を表す値
- 型番
 - メーカー毎に付与される機器の型番
- メーカーコード
 - 機器を製造した会社名を表す値
- 機種名
 - メーカー毎に付与される機器のブランド名やシリーズ名を表す値

接続構成情報は、NW機器が保持する情報であるForwarding Data Base(FDB)と同義である。NW機器におけるポートと、そのポートに接続されたエンド端末のMACアドレス、或いは他のNW機器のMACアドレスが対になった情報である。なお、この規格ではL2AgentとL3Agentが情報を送信するプロトコルについて規定されており、Managerの機能については詳しく記述されていない。

3. トポロジの検出

本章では、HTIPにより収集された接続構成情報であるFDBからリンクレイヤのトポロジを検出するためのアルゴリズムについて述べる。

3.1 前提条件

3.1.1 対象とするネットワーク環境

管理するネットワークは、一つのリンクレイヤブロードキャストドメインを対象とする。図2に示すように、一つのリンクレイヤブロードキャストドメイン外の非IPネットワークや、L3で分離されたIPネットワークについては対象外とする。また、ホームゲートウェイ(HGW)のFDBにはホームネットワークのWANに当たる情報を含んでい

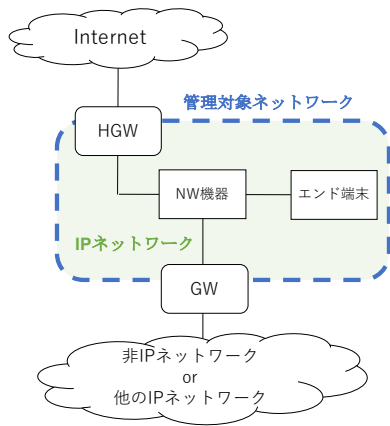


図2 対象のネットワーク環境
 Fig. 2 Target network environment

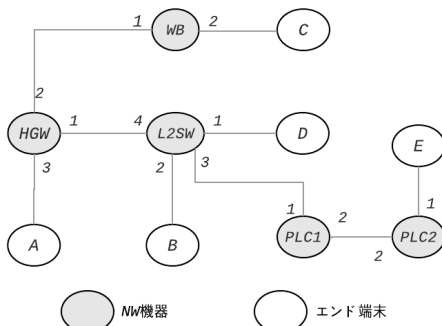


図3 図1のグラフ表現
 Fig. 3 The network graph of figure 1

ないこととする。

NW機器の全FDBを一つの機器(以下、Manager)で収集し、このManagerにてネットワークのトポロジを検出することを想定している。一般的にはHGWがManagerとして動作することが多いため、以下においてもHGWをManagerとする。Managerは、HTIPを用いてFDBを収集し、リンクレイヤのトポロジを検出する。また、HTIPではNW機器はFDBが含まれたLLDPフレームを定期的にブロードキャストする。

3.1.2 グラフ表現

ネットワークは、各機器をノード、接続媒体をエッジで表現することでグラフとして図3のように表現することができる。図3は図1のネットワークのグラフ表現である。グラフの丸の中のラベルはその端末のMACアドレスを示している。より簡易化するため、これらのラベルはノードの名前とする。例えば、図3内のノードDは図1内のPCであり、PCのMACアドレスはDである。また、NW機器については灰色の丸で表現されており、Wireless BridgeをWR、L2 SwitchをL2SW、PLC Gateway1をPLC1として、名前を省略している。NW機器から伸びるエッジに記載された数字はポート番号を示している。

表1 図3のネットワークに対応したFDB

Table 1 The network graph of fig. 1

HGW		L2SW	
port	MAC Address	port	MAC Address
1	L2SW, PLC1, PLC2, B, D, E	1	D
2	WB, C	2	B
3	A	3	PLC1, PLC2, E
		4	HGW, WB, A, C

WB	
port	MAC Address
1	HGW, L2SW, PLC1, PLC2, A, B, D, E
2	C

PLC1	
port	MAC Address
1	HGW, L2SW, WB, A, B, C, D
2	PLC2, E

PLC2	
port	MAC Address
1	E
2	HGW, L2SW, WB, PLC1, A, B, C, D

3.1.3 FDB

表1は、図3のNW機器のFDBを示している。なお、一つのFDBは一つのNW機器におけるFDBである。

3.2 トポロジ検出アルゴリズム

SNMPを用いてNW機器から収集したFDBからリンクレイヤのトポロジを検出する研究が存在する[2][3]。ネットワークのトポロジの検出にあたり、NW機器同士が一つのNW機器を介してでも接続されている場合を「接続」、一つの物理線で接続されている場合を「直接接続」と定義している。また、トポロジの検出に用いる条件式内の表記法について以下のように定義されている。

- C_{ip} : NW機器 i のポート p に接続されているNW機器の集合
- D_{jq} : NW機器 j のポート q に直接接続されているNW機器

直接接続を検出するため、以下の三つの条件式が示されている。

- 条件1: $C_{ip} = \{j\} \Rightarrow D_{ip} = j$
- 条件2: $(D_{ip} = j) \wedge (C_{jq} \ni i) \Rightarrow D_{jq} = i$
- 条件3: $(C_{ip} \ni j) \wedge (C_{jq} \ni i) \wedge (C_{ip} \cap C_{jq}) \neq \phi \Rightarrow D_{ip} \neq j$

ネットワークのトポロジを検出するためには、NW機器同士が直接接続されているかどうかを決定する必要がある。まずNW機器間の接続の関係については、表1から抽出することが可能である。表2に図3のNW機器間の接続関係を示す。

NW機器間の直接接続を決定するには、以下の(1)~(3)を新たな直接接続を決定できなくなるまで繰り返す。

- (1) 条件1を表2に適用し、各列の中で唯一に決まるポートを選択し、直接接続を決定する。
- (2) (1)の結果について条件2を適用する。

表 2 NW 機器間の接続関係

Table 2 Connection relationship between NW devices

		Connection Src				
		HGW	L2SW	WB	PLC1	PLC2
Connection Dst	HGW	-	4	1	1	2
	L2SW	1	-	1	1	2
	WB	2	4	-	1	2
	PLC1	1	3	1	-	2
	PLC2	1	3	1	2	-

表 3 NW 機器間の直接接続関係

Table 3 Direct connection relationship between NW devices

Connection Src		Connection Dst
Device Category	Port Number	Device Category
HGW	2	WB
WB	1	HGW
PLC1	2	PLC2
PLC2	2	PLC1
HGW	1	L2SW
L2SW	4	HGW
L2SW	3	PLC1
PLC1	1	L2SW

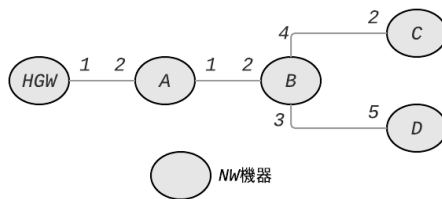


図 4 ネットワークトポロジ例

Fig. 4 An example of network topology

(3) (2)(3)の結果から同一ポートに接続される他のNW機器は直接接続ではない。

この結果から表3のNW機器間の直接接続関係を得る。

条件1, 2で直接接続を決定できない場合がある。図4に示すNW機器のみからなるトポロジについて考える。NW機器Aから情報を完全に得られない場合、HGWとNW機器Bとが直接接続していることになり、NW機器Aの存在がわからなくなるだけである。一方で、NW機器Bから情報が完全に得られない場合、条件1を用いてNW機器A, C間の直接接続を決定するだけで、NW機器A, C, D間の直接接続を決定することはできない。これは、NW機器間A, C, D間で接続媒体を共有しているようなネットワークでも同様のことが言える。そこで、条件3を用いることで、条件1, 2だけでは検出できなかった直接接続を決定できる場合がある。

4. トポロジ情報の記述

本章では、検出されたトポロジ情報を記述するための記述モデルについて検討を行い、ホームネットワークにおけるトポロジ情報を記述する記述モデルを提案する。

4.1 既存ネットワーク記述モデル

情報を構造化することでシステムにおけるデータの交換を可能とし、利便性を向上させることを情報のモデル化という。本節では、既存のネットワーク記述モデルのうち、トポロジ情報の記述に特化した関連技術を述べる。

4.1.1 YANG data model for Network Topology

YANG data model for Network Topologyとは、YANG(Yet Another Next Generation)というデータモデルでネットワークトポロジを記述することを目的としたネットワーク記述モデルである[4]。YANGは、ネットワーク管理機器の管理プロトコルであるNETCONF(Network Configuration Protocol)で使用される設定項目や状態などの構造を人が解釈しやすい形式で記述することを目的としたデータモデル言語である[5][6]。

YANG data model for Network Topologyのデータ構造は、ネットワークトポロジの基本となるノードとリンク、終端点が定義されており、属性を付加していくことで様々なトポロジの表現が可能である。

4.1.2 NetJSON

NetJSONとは、L2およびL3ネットワークの基本的な要素を記述するために設計されたJSON(JavaScript Object Notation)に基づくネットワーク記述モデルである[7]。JSONは、軽量なデータ記述言語の一つである[8]。構文はJavaScriptにおけるオブジェクトの表記法を基本としているが、JavaScript専用のデータ形式ではなく、様々なソフトウェアやプログラミング言語間のデータの交換に使えるよう設計されている。

NetJSONでは、以下の五つのデータ記述モデルが定義されている。

- NetworkRoutes
- DeviceConfiguration
- DeviceMonitoring
- NetworkCollection
- NetworkGraph

このうち、NetworkGraphは、ネットワークのノードとリンクをリストとして定義しており、ネットワークトポロジの可視化を目的として開発された記述モデルである。

4.2 ネットワーク記述モデルの検討

YANG data model for Network TopologyとNetJSONの特徴について比較して検討する。NetJSONはシンプルな記述モデルであるが、YANG data model for Network Topologyに比べて拡張性に乏しいという面がある。また、YANG data model for Network TopologyはYANGというデータモデルに基づいているため、YANGモデルとして定義することで、JSONやXMLなどの様々な形式への変換が可能になることからNetJSONに比べて汎用性が高いという面もある。そこで、本研究では、YANG data model

```

module: http-topology
augment /topo:network-topology/topo:topology/topo:node:
  +-rw mac?          yang:mac-address
  +-rw DeviceCategory? string
  +-rw ManufacturerOUI? string
  +-rw modelName?   string
  +-rw modelNumber? string
augment /topo:network-topology/topo:topology/topo:link:
  +-rw ifType?   iana:iana-if-type
augment /topo:network-topology/topo:topology/topo:link/topo:source:
  +-rw port?   inet:port-number
augment /topo:network-topology/topo:topology/topo:link/topo:destination:
  +-rw port?   inet:port-number
  
```

図 5 ホームネットワークトポロジ対応のスキーマツリー

Fig. 5 schema tree for home network topology

for Network Topology 及び Net.JSON, それぞれのネットワーク記述モデルを基本として, 記述モデルを拡張する形でホームネットワークのトポロジ情報が記述可能なモデルを提案する.

4.3 ホームネットワークにおけるネットワーク記述モデル

ホームネットワークのトポロジを把握するには, 少なくとも Chassis ID と機器情報 (区分, 型番, メーカーコード, 機種名), 接続構成情報 (ポート番号, ポート種別, ノードの接続関係) を記述可能なモデルにする必要がある. 本節では, YANG data model for Network Topology 及び Net.JSON について, ホームネットワークのトポロジの把握を可能とするネットワーク記述モデルについて述べる.

4.3.1 ホームネットワークトポロジ対応

YANG data model for Network Topology

YANG は既に定義されているモデルにスキーマツリーを接ぎ木することができる. 図 5 に HTIP で得られるトポロジ情報を記述するための YANG data model for Network Topology に接ぎ木するスキーマツリーを示す.

4.3.2 ホームネットワークトポロジ対応 NetJSON

Net.JSON で定義されている NetworkGraph に基づいてホームネットワークのトポロジ情報を記述する. Net.JSON の NetworkGraph は, オプションを含めることが可能である. このオプション部分にホームネットワークのトポロジ情報である機器情報などの情報を記述する.

5. ホームネットワークトポロジ検出システム

本章では, 提案するホームネットワークのトポロジを自動的に検出した後, 適切な形式で記述し格納する手法について述べる.

5.1 提案手法

提案するトポロジ検出システムの動作例を図 6 に示す. 図中の管理運用情報統合のデータベース (DB) は, トポロジ情報のみならず障害情報など, HTIP や SNMP などのプロトコルから得られる管理運用情報を統合するためのシステムであり, 同研究室で開発が進められている. 本研究では, 橙色の部分に重点を置いている. なお, 以下の番号は図 6 中の番号と対応している.

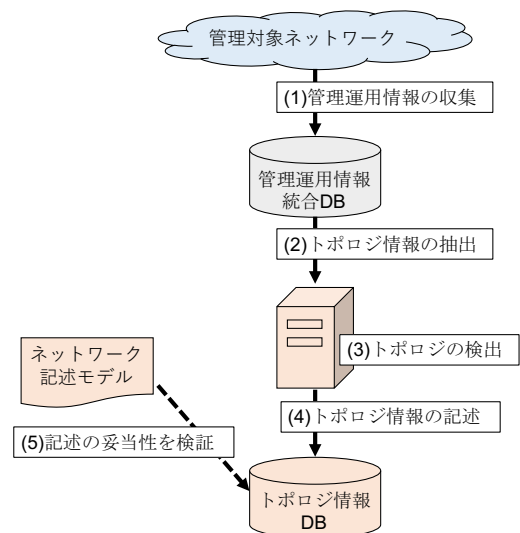


図 6 トポロジ検出システムの概要

Fig. 6 Structure of topology discovery system

(1) 管理運用情報の収集

- 管理対象のホームネットワークから HTIP などの管理運用プロトコルを用いて情報を一箇所 (管理運用情報統合 DB) に集約する.

(2) トポロジ情報の抽出

- 管理運用情報統合 DB から管理対象のホームネットワークトポロジを把握するために必要なトポロジ情報を抽出する.

(3) トポロジの検出

- 抽出されたトポロジ情報のうち接続構成情報 (FDB) を使用してトポロジ検出アルゴリズムを実行し, トポロジを検出する.

(4) トポロジ情報の記述

- 検出したトポロジ情報を作成したネットワーク記述モデルに即して記述する.

(5) 記述の妥当性の検証

- ネットワーク記述モデルをスキーマ (例:JSON スキーマ) として定義し, 既存のバリデータを用いて検証する.

5.2 プロトタイプシステムの実装

トポロジ検出システムのプロトタイプとして, 収集されたトポロジ情報からトポロジを検出した後, 提案するネットワーク記述モデルに基づいてトポロジ情報を格納し, それらを可視化するアプリケーションを作成した. データ記述言語には JSON を使用した. 図 7 にトポロジビューアのシステム概要を示す. また, 図 8 にトポロジビューアの動作イメージを示す.

ここでは, 図 3 に示したネットワークのトポロジ情報を入力としている. トポロジを描画する前処理として, トポロジ情報のうち FDB の情報からトポロジ検出アルゴリ

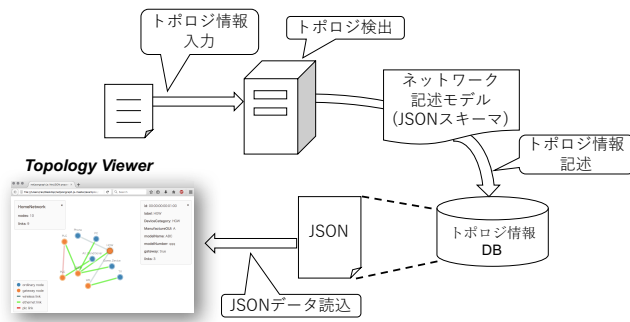


図 7 トポロジビューアのシステム概要
Fig. 7 Topology Viewer system

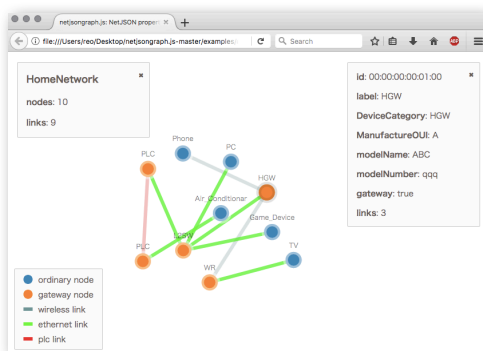


図 8 トポロジビューア
Fig. 8 Topology Viewer

ズムを用いてトポロジを検出している。その後、提案するホームネットワークに対応した記述モデルに基づいた、JSON データを生成している。なお、今回使用したネットワーク記述モデルは NetJSON を拡張したモデルである。最後に、ウェブブラウザ上で動的コンテンツを描画する JavaScript ライブラリ D3.js を用いてトポロジを可視化している。図 8 はノードとリンクの種別を色で表示している。また、ノードをクリックすることで、その機器の情報である機器名や型番などを一覧で表示することができる。トポロジ検出システムが保持しているトポロジ情報をいかにユーザに提示するかは、今後の検討課題の一つである。

6. まとめと今後の課題

ホームネットワークでは、異なるネットワークや多種多様な機器が接続されることで、ホームネットワークのトポロジが複雑化している。これに対し、ホームネットワークに接続された各機器より収集された接続構成情報からトポロジを自動的に検出し、ネットワーク記述モデルによってトポロジ情報を記述する手法を提案した。これにより、多岐にわたる端末間、或いはネットワーク技術間のトポロジをレイヤ横断的に記述することができ、ホームネットワークのトポロジ全体を把握することを可能とする。また、ネットワーク記述モデルを利用することにより、トポロジ情報を計算機で読み込み可能な形式となる。これによ

り、ホームネットワークの障害原因の特定を容易化・自動化することが可能になり、ホームネットワークの可用性の向上に貢献できる。

今後の課題として、トポロジを自動的に検出し、形式的に記述して可視化する手法をプロトタイプシステムとして実装した。しかし、システムで利用している接続構成情報はあらかじめ用意した情報であり、HTIP が実装された実際のホームネットワーク環境、或いはシミュレーションされた環境において実験を行う必要があると考えている。

参考文献

- [1] JJ-300.00 ホーム NW 接続構成特定プロトコル第 3.0 版, 2017. http://www.ttc.or.jp/jp/document_list/pdf/j/STD/JJ-300.00v3.pdf
- [2] 吉田和幸, ネットワークトポロジの発見とその表示について, 大学情報システム環境研究, VOL.10, pp68-74, 2007.
- [3] 河野優, 釜崎正吾, 平川龍, 大浦昇, 吉田和幸, Layer2 ネットワーク構成情報の検出アルゴリズムの改良について, 情報処理学会「分散システム/インターネット運用技術シンポジウム 2005」論文集 pp. 61-66, 2005.
- [4] A.Clemm, J.Medved, R.Varga, N.Bachadur, H.Ananthkrishnan and X.Liu: A Data Model for Network Topologies, Work in Progress, IETF, 2017. <https://www.ietf.org/id/draft-ietf-i2rs-yang-network-topo-20>
- [5] M. Bjorklund: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), RFC6020, IETF, 2010.
- [6] M.Bjorklund: The YANG 1.1 Data Modeling Language, RFC7950, IETF, 2016.
- [7] NetJSON: data interchange format for networks, 2015. <http://netjson.org/rfc.html>
- [8] T.Bray: The JavaScript Object Notation (JSON) Data Interchange Format, RFC8259, IETF, 2017.