6ZA-01

制御ネットワーク向けネットワーク検証基盤の提案

鈴木　智紀†　　向井　宏明†
金沢工業大学†

1. はじめに

近年，工場等の制御ネットワークをインターネットに接続する必要性が増加している．これにより，セキュリティ上のリスクも増加している．

しかし，制御ネットワークに適した検証基盤は少ない．また，制御ネットワークには，Ethernet 以外の規格や独自のプロトコルも多く存在する．

本稿では，独自プロトコルの検証やネットワークの運用検証，セキュリティの検証作業が容易に行える，制御ネットワークに適した検証基盤を提案する．

2. 既存手法

2.1. 関連技術

ネットワークシミュレータは，数多く存在している．代表的なものとして，Cisco Packet Tracer[1]や GNS3[2] ，ns-3[3]などが挙げられる．

Cisco Packet Tracer は，Cisco が提供しているネットワークシミュレーションツールである．ネットワークの学習をサポートするものである．しかし，Cisco が提供するネットワーク機器のみとなるため，その他のメーカーのものを検証することは難しい．

GNS3 は，ハードウェアエミュレートを行っている．そのため，実際のネットワーク OS を利用することが出来る．また，Appliances にてネットワーク機器以外の機能も追加されている．そのため，ネットワーク機器のみの検証だけではなく，周りも含めた検証も行える．しかし，実際のルーター機器を検証する際には，困難が伴うことが予想される．

ns-3 は，シナリオに沿った検証ができるシミュレーターである．

A Proposal of network verification platform for control networks
† Tomonori Suzuki, Hiroaki Mukai
Kanazawa Institute of Technology

2.2. 問題点

制御ネットワークの問題点を２点挙げる．

1 点目は，工場等のネットワークを構成するネットワーク機器の多くは，スイッチなどのレイヤー2 の機器を多数利用し構成されている．また，工場に関連するネットワークには，情報系ネットワークだけではなく，ネットワーク末端のフィールドネットワークもある．そこでは，PC などの情報端末ではなく，PLC などの機器が接続されていることもある．そのため，これまでのネットワークシミュレータのネットワーク機器の設定を検証するのではなく，工場機器も含めた検証が必要となる．

2 点目は，制御ネットワークにおけるプロトコルには独自プロトコルも多く存在することである．そのため，ネットワークシミュレータ内で利用するには，物理機器との接続が必要になる場合や，機器を模擬したものを作成し検証を行う必要がある．場合にもよるが，模擬したものを作成した場合には，ほかの検証環境に移動させた場合に動作しなくなってしまうことが想定される．そのため，検証環境が変わったとしても，動作するものが必要である．

3. システムの概要

前章で述べた問題点を解決するため，制御ネットワーク向けの検証基盤を提案する．制御ネットワークの例として，一般的な工場の構成を図１に示す．各ゾーンにそれぞれの役割を持ったゾーンが存在する．

まず，1 つ目のスイッチなどのネットワーク機器が多く存在する問題についてである．制御ネットワークでの検証作業は，ネットワーク機器の設定の検証だけではなく，工場システム全体の動作検証という意味合いも大きいと考える．工場機器だけではなく，IT システムとの連携も確認する必要がある．そのため，よりレイヤーの低いシミュレーターの提案が必要である．また，近年工場での利用が進んでいる無線機能の追加も必要な項目であると考える．

2 つ目での独自機器のプロトコルは，検証を行

うものやその機器を販売する組織が何らかの方法で共通の検証基盤を利用する必要がある．そのため，今回は，コンテナを利用しこれを解決する．近年，利用が進んでいるコンテナ技術は，ハードウェアなどの環境と切り離すことが可能であるため，検証環境の差異を埋めることが可能である．

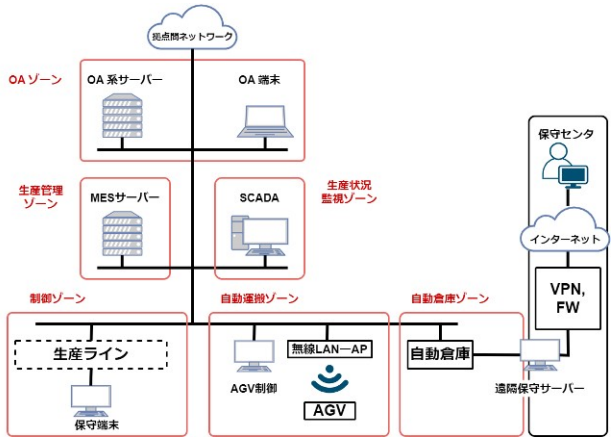

図１　想定する工場の構成

4. 提案システム

提案する制御ネットワーク向けネットワーク検証基盤を図 2 に示す．ネットワーク検証基盤は，トポロジー生成，環境構築，可視化で構成される．

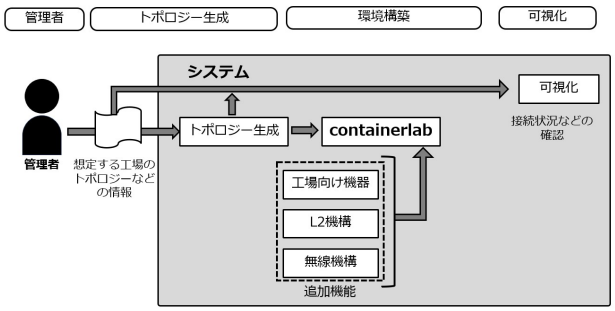

図２　提案システム

4.1.　トポロジー生成

トポロジー生成段階は，大規模なネットワークを生成する際は有効な機能である．機能として，木構造のトポロジーを生成するものとなっている．また，工場で利用されるコンテナイメージを末端に配置し，工場でのゾーンが分けられるように生成する．トポロジー生成は，あくまで補助的な機能である．検証において，現実に沿ったトポロジーが必要ある．

4.2.　環境構築

containerlab[4]を利用し構築する．

containerlab とは，コンテナベースのネットワークトポロジーを作成・管理するツールである．コンテナオーケストレーションツールは，ユーザーが定義するネットワークを簡単に接続するには向いていない．containerlab は，ネットワーク接続を調整および管理をするための環境を提供している．containerlab 環境構築には，通常 yaml ファイルを利用し，ノードの種類，接続先を記述する．

containerlab 選定した理由は，コンテナを利用することで，検証環境の差異を少なくさせることが目的である．

また，containerlab には，Cisco や Juniper などのネットワークベンダーのネットワーク OS がコンテナ化されたものがすでに提供されている．そのため，新たに自身でネットワーク機器を準備する必要がない．

コンテナを自身で作成することで，検証することも出来るため，作成したコンテナイメージの再利用が可能となるためである．

4.3.　可視化

可視化機能では，利用者が作成した工場のトポロジーなどの情報をもとに，ノード間の接続情報について視覚的に確認できる機能を追加する．

5. おわりに

本稿では，制御ネットワーク向けネットワーク検証基盤の提案を行った．

今後の展望は，対応するプロトコルの拡大や複雑な条件に対応したトポロジー生成，通信状況のさらなる可視化に取り組んでいきたい．

参考文献

[1] Cisco Packet Tracer, https://www.netacad.com/ja/courses/packet-tracer, (参照:2024.1.11)
[2] GNS3, https://www.gns3.com/, (参照:2024.1.11)
[3] ns-3, https://www.nsnam.org/, (参照:2024.1.11)
[4] containerlab, https://containerlab.dev/,(参照:2024.1.11)
[5] 産業サイバーセキュリティ研究会 ワーキンググループ 1(制度・技術・標準化), 「工場システムにおける サイバー・フィジカル・セキュリティ対策 ガイドライン」,(2023.11.16)