知識ベースに基づいたネットワークトラブルシューティング自動化手法

紅林輝† 河口信夫†
†名古屋大学大学院工学研究科

1. はじめに

ネットワークにトラブルが発生し、ネットワークの利用が長時間困難になると、利用している一般ユーザに大きな迷惑をかけてしまうため、トラブルシューティングはネットワーク管理者の重要な仕事の一つである。ネットワーク管理者育成の場では、トラブルシューティングに関する教育も大切だと考えられており、それを支援する仕組みも提示されている[2]。

ただし、ベテランの管理者であってもトラブルの原因を絞込むには大変な労力を要する場合もあり、トラブルシューティングはネットワーク管理者の負担を大きくする要因の一つとなっている。

本研究は、ネットワークのトラブルの原因を自動推定する仕組みを提供し、ネットワーク管理者の負担の軽減を目指している。本稿では、検討している手法とシステム、及び進捗状況を述べる。

2. アプローチ

人工知能等の分野で研究が進められてきた知識ベースを、ネットワーク構築・運用に利用する研究が、これまでにいくつかなされている。

例えば、文献[3]では、ユーザの要求に応じて適切なネットワークのサービスの構成や制御をシステムが行い、そのユーザの実行環境を考慮したネットワークの構成を自動的に行う、といったシステムが提案されている。

また、トラブルシューティングに関して、障害情報に応じて、管理者が打ち込む必要があるコマンドをあらかじめシステムにスクリプトとして記述しておき、障害を検知するとそれを実行し、トラブルチケットシステムと組み合わせることによって普及作業の効率化を図る、という研究もなされている[4]。ただし、最終的に原因を絞りこむためには、コマンドの実行結果をもとに、管理者が頭の中で原因となり得る項目を考える必要がある。もし、トラブルの原因を管理者に提示できれば、トラブルシューティングの負担を、より軽減できる。

本研究では、あらかじめトラブルの原因とそれによって引き起こされるトラブルの影響範囲との依存関係を知識ベースとして記述しておく。その知識ベースを元に、トラブルが起こっている状態と、トラブルが起きていない状態、それぞれの情報を用いてトラブルの原因を推定し、管理者に提示する。

2.1. 原因推定のための情報の取得

トラブルの原因推定に必要な情報として、ネットワークの構成等の情報はあらかじめシステムに入力しておかなければならない。ただし、トラブルの原因をより正確に推定するためには、ネットワークを利用している一般ユーザの端末の接続状況やトラフィック等の情報も必要である。これらの情報は、その時々にネットワーク機器から取得しなければ把握が難しい。さらに、大規模なネットワークでは、多数のネットワーク機器からこれらの情報を得るだけでも大変な作業となる。

現在、名古屋大学を中心として、新たなネットワーク管理システムを構築するプロジェクトが進んでいる[1]。来年度に行われる名古屋大学のキャンパスネットワークの更新に合わせて導入出来るように準備が進んでいる。このシステムは、運用管理に必要な機能の網羅をコンセプトの一つとして、設計・開発が行われている。その中には、SNMP によるネットワークの監視や、ネットワーク機器の config 情報の管理といった、本研究に必要な情報の管理も含まれる。このシステムでは、API 等によりそうした情報を提供できるように構想が進んでいる。

本研究では、このシステムを利用して、原因推定に必要な情報を取得することを想定する。したがって本研究では、トラブルの原因推定に必要な情報は取得出来ることを前提とし、いかにトラブルの原因を推定するか、という事に焦点を当てる。

2.2. 知識ベースと Prolog

前述したように、本研究では、トラブルの原因とその影響範囲との依存関係を知識ベースとして記述する。例えば、「ケーブルが切れている」という原因と「そのケーブルを挟んでいる機器同士は通信出来なくなる」という影響範囲を依存関係として記述しておく。こうした依存関係を記述しておき、障害情報から原因を絞り込む。

知識ベースにおけるルールの記述について、論理型言語である Prolog を用いて記述が出来ないかと、取り組み中である。

Prolog では、

IF A1,A2 THEN B(A1 かつ A2 が成り立てば B)

というプロダクションルールを次のような

定義節で表現する。

B :- A1,A2.

Prolog プログラムでは、事実や規則をすべてこのような定義節で表すため、トラブルとその原因との依存関係もこのようなルールで記述する。例えば、「ケーブルであれば、切れて通信出来なくなる可能性がある」というルールは、次のように表現できる。大文字のアルファベットは変数である。

```
line(cable_1).
line(cable_2).
…
failure(X,cut) :- line(X).
```

このように書かれたプロダクションルールの集まりを Prolog プログラムで記述し、Prolog インタプリタが後ろ向き推論方式にしたがって、実行する。

実行例を、以下に示す。「pc1 から外のネットワークに出られない」という障害情報を入れて実行すると、以下のように原因 A が列挙されている。以下は簡単な例であるが、端末が壊れている可能性がある、ケーブルが切れている可能性がある等が表示されている。

```
?- fail_link(pc1,wan,A).

A = fail(pc1, hard_fail) ;
A = fail(port_b2, hard_fail) ;
A = fail(port_b1, hard_fail) ;
A = fail(cable_4, cut) ;
```

2. 3. システムの構想

本システムの最終的なイメージ図を、図 1 に示す。

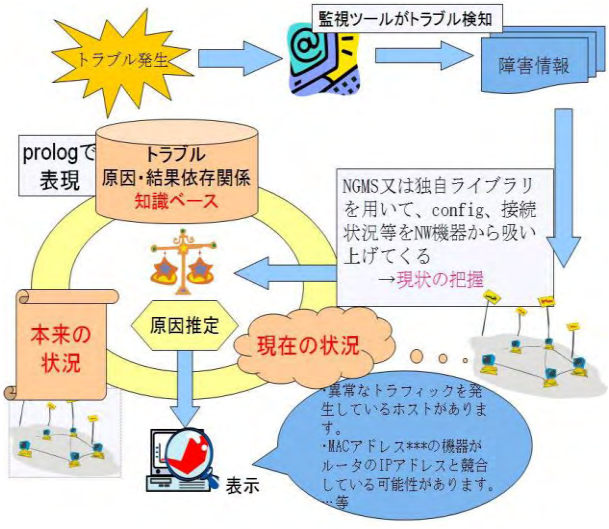

図 1 システムイメージ

最初に、監視サーバあるいは人間が、トラブルが起きたことを検知したら、提案システムに障害情報を入力する。

次に、システムは、障害情報が入力されたら、前述したように必要な情報を NGMS から取得する。ここで得た情報が、トラブルが起きている状態の情報であり、これに加えて、トラブル検知前の状態の情報があればそれを正常な状態の情報として、より正確なトラブルの原因推定に利用することが出来る。取得した情報から、2 章で述べた知識ベースをもとにトラブルの原因を推定する。

最後に、推定された原因を管理者に提示する。

3. おわりに

以上、述べたように現在、知識ベースの構築に取り組んでいる。しかし、まだ単純なネットワークのものしか記述していないため、現実のネットワークの構成に則した複雑な依存関係も記述する必要がある。

今後の取り組みで、知識ベースからトラブルの原因を推定する部分を実現し、ネットワーク管理支援システムとして実装させたい。

参考

[1]NGMS http://ngms.info/
[2]立岩祐一郎,安田孝美,横井茂樹：仮想環境ソフトウェアに基づく Linux ネットワークトラブルシューティング実習環境提供システムの開発,情報処理学会研究報告,コンピュータと教育研究会報告,2007(123),pp.37-44,2007 年
[3]西川健一他：知識ベースを用いたユーザネットワーク運用支援システムの検討,電子情報通信学会技術研究報告,情報ネットワーク,105(113),pp.29-34,2005 年
[4]岡山聖彦,山口英,宮原秀雄：作業のスクリプト記述に基づいたネットワーク管理支援システム SPLICE/NM の設計と実装,電子情報通信学会論文誌,vol.J81-D-1,No.8,pp.1014-1023,1998 年