第 2 章
トラブルシューティングの方法論

ネットワーク管理者の責任は、「パフォーマンスの最大化」「可用性の最大化」「コストの最小化」「修理時間の最小化」の 4 つに集約されます。

この章では、「修理時間の最小化」について説明します。機能を回復させるための時間は、「準備」と「技法」の 2 つの要因に左右されます。前章では、「準備」、つまり、文書化や定期的な予防保守などの要因について説明しました。この章では、ダウンタイムの最小化のために利用できる「技法」について説明します。

この章で説明するトラブルシューティングの各実践方法は、適切な文書が存在し、適切なツールが使用できることを前提としています。必要な準備がなされていない場合、トラブルシューティングははるかに手間と時間のかかるものになります。

注:
シスコのトラブルシューティング テストは、特定のアプローチを想定しているものではありません。特定の状況では、さまざまに異なるアプローチが成功する場合があります。このテストは、科学的手法に基づいて、トラブルシューティングへの体系的なアプローチを推奨するものです。

原理

通常、科学的手法は、次の 6 段階のプロセスとして記述されます。

1. 問題の定義
2. 情報の収集
3. 仮説の立案
4. 仮説のテスト

5. テストの分析
6. 結果の解釈（および、必要な場合は新しい仮説の立案）

第 1 段階（問題の説明）は、通常、ユーザが問題を報告する際に行われます。最初の問題の説明は、「インターネットがダウンした」などのように、曖昧であったり、非常に一般的であったりする傾向があります 。したがって、トラブルシューティング担当者の最初の対応は、より多くの情報を収集し、より明確に説明することです 。ユーザと話をしたり、個人的な意見を聞いたり、Netflow や syslog、SNMP モニタなどの管理システムを参照したりすることで、症状を判断できます。

問題の適切な説明が得られたら、仮説を立案できるようになります。仮説とは、症状が類似している仮定の問題を指します。仮説は、通常、それ自体を証明または反証する方法を提示します。たとえば、WAN 接続がダウンしていることが疑われる場合、インターフェイスのステータスを確認したり、リモート デバイスに ping を実行したりすることによって、その仮説をテストします。

テスト結果によって、仮説を支持または反証します。1 つのテスト結果では仮説の証明はできず、支持のみが可能です。たとえば、ping は WAN 接続のテストに使用できる可能性があります。ping のタイムアウトは、それだけでは、決定的とはみなされません。ターゲットがシャットダウンしていることや、あるいは、ICMP をドロップしているファイアウォールが存在していることも考えられます。テスト結果は、さまざまに異なるエビデンスによって確認するべきです。テスト結果が仮説と矛盾している場合は、新しい仮説を立てます。

仮説が妥当な説明として受け入れられる場合は、問題を解決するためのアクションを起こします。当然ながら、あらゆるアクションは別の種類のテストとなります。このアクションで問題が解決されなかった場合は、新しい仮説を立案して、プロセスを繰り返します。

体系的なトラブルシューティング

「*体系的なトラブルシューティング*」とは、情報を収集し、仮説を立案し、テストを実施する体系的な方法を示すものです。体系的なアプローチでは、あるテストに失敗した場合、考えられる解決方法クラス全体を除外し、円滑に次の仮説を提示します 。非体系的（ランダム）なアプローチでは、通常、より多くの時間がかかり、成功する確率も低くなります。

成功する技法にはさまざまなものがありますが、それらに共通している特徴は、データを収集して分析する綿密で徹底的なアプローチです。

- **トップダウン方式**：OSI アプリケーション層から開始し、下位の層に移動します。
- **ボトムアップ方式**：OSI 物理層から開始し、上位の層に移動します。
- **分割統治法**：ネットワーク層から開始し、エビデンスをたどって、各仮説別のテストを作成します。
- **パス追跡方式**：「パケットの観点」を考慮し、パケットがネットワーク移動時に通過するデバイスやプロセスを調査します。これを実行するには、各デバイス内の動作の順序を理解する必要があります。
- **相違特定方式**：古いバージョンや類似デバイスと設定を比較します。Diff や WinDiff などのツールを使用すれば、こうした比較を簡単に行うことができます。

7	アプリケーション
6	プレゼンテーション
5	セッション
4	トランスポート
3	ネットワーク
2	データ リンク
1	物理

図 2-1
OSI モデル

- **問題の移行**：コンポーネントを交換して、問題がデバイスとともに移行するかどうかを確認します。

唯一の「最適な方法」はありませんが、技術者が特定の問題に対して、より直感的な方法やより適切な方法を見つけ出すことができる場合があります。各技法をより良く理解し、必要に応じてアプローチを変更することを推奨します。2 つのトラブルシューティング方法には、特別な注意が必要です。ほとんどの技術者は、豊富な経験を積んでおり、特定の問題の解決方法を直感的に見つけ出すことができます 。これがうまくいけば非常に印象的ですが、大切なのは、うまくいかないときに無作為に策を講じないことです。

また、ネットワーク使用者は、ほぼ同時に起きたことを見つけ出そうとします。これは、同じタイミングで起きた出来事に因果関係があるとみなす考えに基づいています。これは論理的に誤っており、「*前後即因果の誤謬*」と呼ばれています。これが問題を解く鍵になる場合もありますが、大規模ネットワークでは多くのことが同時に発生します。このトラブルシューティング方法では、誤った方向に向かいがちです。

トラブルシューティング方法

ネットワークのトラブルシューティングは、科学的手法を反映するいくつかの段階に分けることができます。

トラブルシューティングの第 1 段階は、問題の定義です。たとえば、ユーザによっては、「電子メールのダウンロードに時間がかかる」場合に、「インターネットがダウンしている」と報告してくる人もいます。また問題を一般化しすぎたり、誇張しすぎたりする場合もあります。ほとんどのユーザは、どの症状が関連しているかを説明する技術的な知識がありません。そのため、トラブルシューティング プロセスではまず、問題の詳しい説明を得ることから開始します。該当するデバイスの名前や場所などの詳細情報を収集するための質問を行います。詳細情報を収集するうえで良い方法の 1 つに、どうしたらその問題が繰り返されるかを尋ねることがあります（Web をブラウズするとこの問題が確認できるか、など）。

問題を定義できたら、その問題に関する情報を収集します。問題の範囲や、影響を受けたその他のデバイスや場所、問題がいつ始まったか、どのようにすれば問題をテストできるか、などです。

情報を収集すれば、1 つまたは複数の仮説を導き出せます。その説を確認または反証するテストを作成し、根本原因を見つけ出します。テストは、ping などの簡単なものから、設定変更の実装などの複雑なものまで多岐にわたり、有効な仮説を切り分けることを目的とします。

テスト プロセスが完了したら、結果を検討します。設定やハードウェアの変更が示唆されているかどうかを確認します。問題が 解決されない場合は、問題の説明や元の仮説を再検討します。問題が完全に解決されず、正確な説明が得られない場合は、仮説は誤りであり、再検討が必要になります。

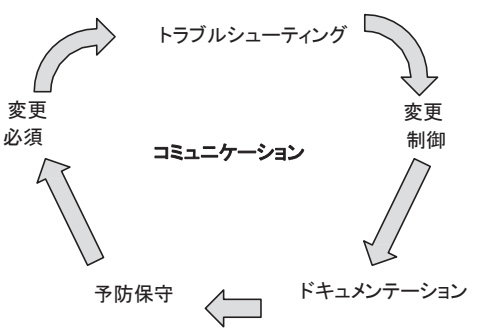
問題が解決されたら、変更を検討します。ネットワークの状態や問題の解決策の伝達、文書の更新が必要にある場合もあります。これまでの明確な段階で、同じ問題がネットワークの別の部分に見られないかどうかを確認します。問題が設定にある場合は、ネットワークで使用されている設定テンプレート全体を検討し、問題の修正を予防的に他のデバイスで繰り返す必要があるかどうかを決定します。

故障/修理サイクルに取り組む場合、組織ごとに独自の方法があります 。ここで重要なのは、論理的かつ系統的に対応し、より大規模なネットワークを完全なものにする機会としてそれぞれの問題をとらえることです。

メンテナンスへのトラブルシューティングの統合

ネットワークの操作はすべて、学習の機会です 。賢明な組織は、学習した情報に基づいて同様の問題を解決し、将来のネットワークの理解に役立てます。変更管理と文書化は、ネットワーク変更によるフィードバックを保守サイクルに取り入れる 2 つの主な方法です（図 2-2 を参照）。

図 2-2
保守サイクル



予防保守は継続的に行われますが、条件の変更や報告された問題により、変更の必要性が生じます。トラブルシューティングにより、ネットワークのアップグレードや修理を行う是正措置を特定します。これらのプロセスの間、問題を把握し、解決方法のフィードバックを収集するために、エンドユーザとの定期的なコミュニケーションが必要です。サイクルを通じて広範囲にエンド ユーザ、チーム内、管理者とのコミュニケーションを行います。