

アプレットを使ったネットワークサービス障害 チェックツール

古里 公彦¹, 鶴 正人², 黒田 英夫¹

¹長崎大学 工学部 電気情報工学科

²長崎大学 総合情報処理センター

あらまし: ネットワークサービスの障害を切り分けるのは、様々な要因があつて難しい。そこで WWW ブラウザでチェック用ページにアクセスするだけで簡単に障害切り分けや管理者への通知ができるようなネットワークサービス障害チェックツールを提案する。これはネットワークの知識がない一般のユーザでも簡単に使え、また JAVA アプレット/アプリケーションの共同で実現されているのでユーザやサーバの機種を選ばず、柔軟性に富んでいる。

Network service health check tool using applet

Kimihiko FURUSATO¹, Masato TSURU², Hideo KURODA¹,

¹ Department of Electrical Engineering & Computer Science, Nagasaki university

² Science Information Center, Nagasaki university

Abstract: It's difficult to trace reasons of network service faults. We introduce a network service health check tool that can automatically trace the reasons of faults and send an indication to the network manager when you access the check page on WWW server. It's so easy that a network beginner can use it, and also it is platform free and flexible because of using JAVA applet/application.

1 はじめに

ネットワークの障害は、切り分けが難しい。この理由として以下のものが考えられる。

- ネットワークは複雑な階層構造を持ち、障害の原因は多様である。
- ネットワークの利用層が拡大し、素人も使うようになった。
- ネットワークサービスの形態がクライアント・サーバ型になって、どちら側の問題かの切り分けが必要になっている。
- 利用者(クライアント)側のハード/ソフトはいろいろある。(マルチベンダー環境)
- 多数の利用者からの問い合わせに対応できるだけのサポート体制がない。
- ネットワークサービスは構成などの変更が多い。

それらを切り分けるには管理者のような専門的知識とツールが必要である。しかし、利用者には前にあげたようにネットワークのことをあまり知らない素人が増加している。そこで利用者側から使えるネットワークサービスチェック(どこまで正常に動いているか)ツールが必要であると考えられる。

このツールを実現する場合、以下の条件が満たされる必要がある。

- 使い方が簡単
- どの利用者の環境でも使える。
- ネットワークサービスの変更にも追従できる。

ネットワーク利用者は少なくとも WWW ブラウザの使い方は知っており、また、多くの WWW ブラウザが JAVA アプレットをサポートしている。そこで、JAVA アプレットによる、ネットワークサービス障害チェック(障害診断)のツールを試作した。

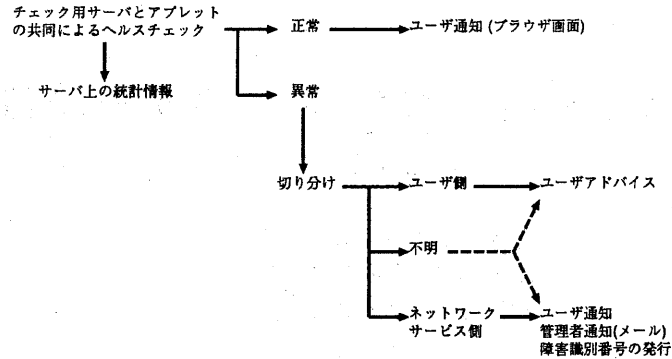


図 1: 基本的な流れ

2 機能概要

試作しているツールはチェック用サーバとアプレットが通信することでヘルスチェックを行う。このとき、サーバ上には統計情報を残す。ヘルスチェックの結果が正常であったならユーザ (ブラウザ画面) に通知する。結果に異常があった場合は障害がなんであるかの切り分けを行う。その障害はユーザ側が原因の場合とネットワークサービス側が原因の場合、また原因不明 (このツールでは切り分けできない) の場合が考えられる。

ユーザ側が原因であると、それをユーザ (ブラウザ画面) にアドバイスをする。ネットワークサービス側が原因であると、それはユーザと管理者に通知をする。この際、障害識別番号の発行をして、障害の処理をやりやすくする。原因不明の場合はユーザと管理者に通知する。図 1 参照。

本ヘルスチェックツールはネットワークの基本的なチェックをする基本チェックページと、各サービスのチェックをするページとがある。

基本チェックページ:

- クライアントとサーバの時刻の表示。これによってサーバ側の時計は正しいと仮定するとクライアントマシンの時計の狂いがわかる。
- 応答時間 (RTT) の表示。これでクライアントとサーバの間の応答時間がわかる。
- ネットワークの経路の表示。これでサーバからクライアント方向の経路がわかる。
- これらの情報を、統計情報としてファイルに残すことで、管理者が、後日これらを検出できる。

- ネームサーバ (DNS) が正常に引けるかどうか調べる。
- 学内 LAN 障害情報のページへのリンク。
- 各サービスのヘルスチェックページへのリンク。

サービスごとのヘルスチェックには、メールサーバ (SMTP、POP)、NNTP (ニュース読みだし)、FTP、Telnet などがあり、それぞれが正常に動いているかどうか等のチェックを行う。

例えば、メールサービスのヘルスチェックページ:

- SMTP サーバのチェック (生死)
- POP サーバのチェック (生死)
- POP サーバの強制復旧
- ネームサーバ (DNS) が正常に引けるかどうか調べる。

各チェックページごとに障害切り分けを行うが、基本チェックページの場合、図 2 のようになる。

3 システム構成

まず、ユーザは図 3 にあるような基本チェックのページがあるサーバ 1 にアクセスする。するとアプレットが読み込まれ動き出す。このアプレットとサーバ側で動く JAVA アプリケーションの間で通信することでクライアントとサーバの時刻、応答時間 (RTT)、ネットワーク経路を得る。ここで、障害が検知された場合、図 2 のような障害切り分けをおこないブラウザに表示する。なお、このサーバでは Solaris2-5-1 上の JDK1.1.4 を使用している。

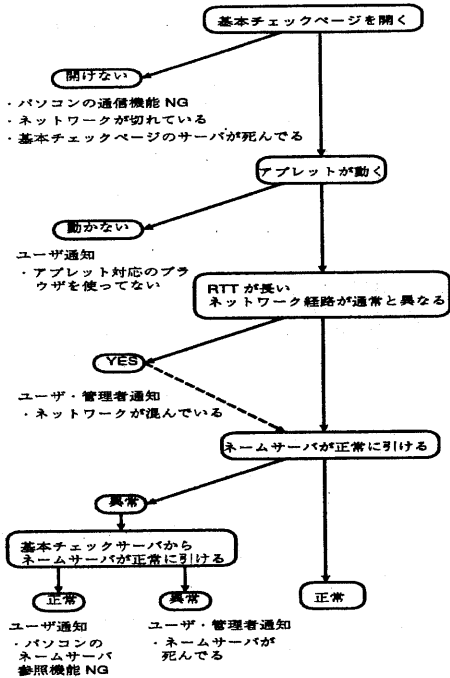


図 2: 基本チェックページでの障害切り分け

また、そのメインページ(図4)には、各サービス(サーバ)へのリンクがはられており、それぞれのサービス(SMTP,POP,NNTP,FTP,Telnetのチェック)(図5)が受けられる。図3では、サーバ1から例えばPOPのチェックを受けようとする、リンクをつたうことでサーバ2へ移り、そこでPOPチェック用アプレットを読んでくることでチェックを行う。

以下メインページでの機能を実現するための方法を示す。

クライアントとサーバの時刻と応答時間はアプレットとサーバ側のアプリケーション間の通信で得ることができる。

ネームサーバのチェックは、アプレットでサーバのドメイン名をクライアント側でIPアドレスに変換できるかで判断する。そのためにエンドユーザには基本チェックページのURLはIPアドレス形式で広報しておく。チェック用アプレットにサーバのドメイン名とアドレスを引数として渡し、クライアント側(アプレット)でサーバのドメイン名をIPアドレス変換してみる。これで得られたIPアドレスとアプレットの引数のIPアドレスとが同じならばネームサーバは正常となる。異常の場合はユーザと

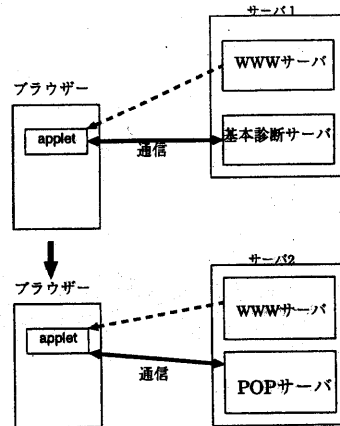


図 3: システム構成

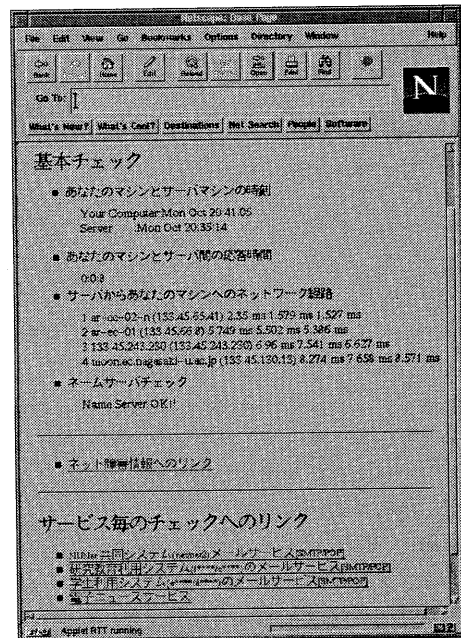


図 4: メインページのイメージ

管理者に通知する。

ネットワークの経路は、サーバからクライアントへの経路を調べる。その方法は、まずクライアント側でIPアドレスを得る。得たIPアドレスをサーバに送りサーバでRuntime.exec()を用いて外部のtracertコマンドを実行し、その結果をクライアントに送ることで表示する。クライアント

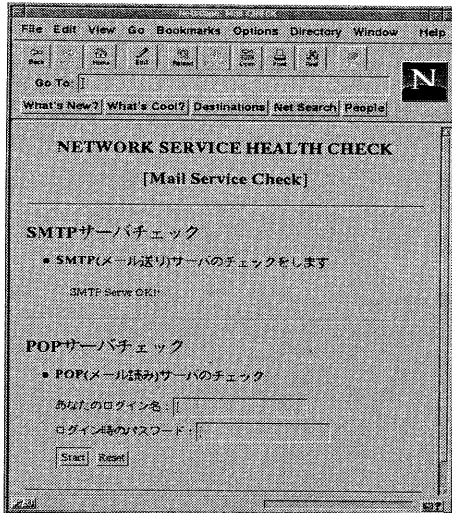


図 5: メールチェックページのイメージ

の IP アドレスを得るために アプレットで `InetAddress.getLocalHost()` メソッドを使っているが、この場合 Netscape Navigator 3.01 では動作するがインターネットエクスプローラ (IE) では正確な IP アドレスを得られない。このため、IE ではネットワーク経路の表示ができないという問題が生じた。IE の場合はユーザに IP アドレスを入力してもらうか、サーバ側で IP アドレスを得るようにしなければならない。

以上メインページで得られるクライアント・サーバ時刻と応答時間 (RTT)、ネットワークの経路などを、サーバ側でファイルに統計情報として残す。これにより、後日、管理者によるチェックなどが行える。現在、統計情報のフォーマットは検討中である。

現在、実装を行っているサービスチェックの中でメールサービス (SMTP、POP) を例にあげる。

SMTP サーバ (メール送) のチェックは SMTP との通信が可能かどうかのチェックを行う。このチェックは RFC821[5] にしたがってメッセージのやり取りを行う。図 6 参照。メッセージのやり取りが終わると SMTP との通信は可能であると判断してアプレットに表示する。もし、以上の通信の途中で SMTP サーバからエラーメッセージがあるとその障害が何であるかを切り分け、表示する。これによりユーザはどんなエラーであるか、知ることができる。

POP サーバ (メール読み出し) のチェックはクライアントと POP サーバとの通信が可能かどうかの

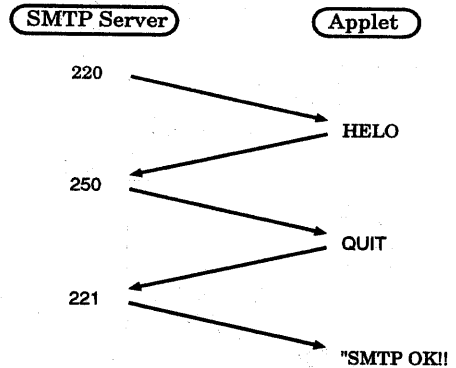


図 6: SMTP サーバとアプレット間の通信 (正常)

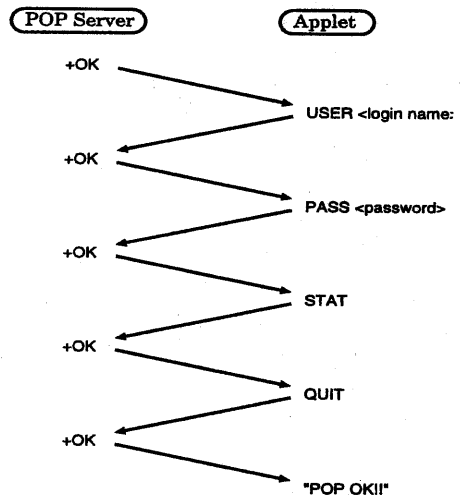


図 7: POP サーバとアプレット間の通信 (正常)

チェックと、POP サーバの強制復旧という機能がある。POP サーバのチェックは、RFC1939[6] にしたがってメッセージのやり取りを行う。POP サーバとの通信の場合ユーザのログイン名とパスワードが必要となるため、これらをユーザーに入力してもらう必要がある。これらをアプレットに取り込むことで図 7 のような通信が可能となる。通信が終わると SMTP の場合と同様な動きをする。

POP の強制復旧には、「前のセッションの強制終了」と「トラブルを引き起こすメールの強制削除 (又は移動)」というのがある。

- 前のセッションの強制終了

一般にセッションの途中でパソコンがハングしても検知できない。そこでサーバはタイムアウト時間で無通信が続くとそのセッションを無効にする。しかし、その場合でもタイムアウト時間がくるまではセッションを開くことができない。長崎大学で導入している POP サーバである YAT server に「前のセッションの強制終了」という合図を送ると、すぐに新しいセッションを開くことができるようになる。このように YAT server に前のセッションを強制終了する合図を送る機能である。

- トラブルを引き起こすメールの強制削除 (又は移動)

ある特定のメール (の中身) がパソコン上のメーラーのバグにマッチしたり、大量のメールが貯りすぎたりして、メーラーがハングしてしまう場合、それらのメールを削除または移動する機能である。

4 まとめ

ネットワークの知識のない一般のユーザも自分でネットワークサービス障害の切り分けを行え、またその情報を自動的に管理者へ通知されるように本ツールを開発した。障害が考えられる時に WWW ブラウザでチェック用ページにアクセスすることで、この切り分けができる。その結果、的確な対策がおこなえるようになると期待される。このように WWW ブラウザでヘルスチェックを行えるので使い方が簡単であり、どの利用者の環境でも使える。

また、クライアントプログラムをアプレットで実現することでクライアントプログラムをサーバプログラムと同じマシン上に保管できる。その結果、クライアントプログラムの配布が不要になり、常に最新バージョンの利用が保証される。つまりネットワークサービスの変更にも追従できる。

現在、設計及び実装中である。このあと長崎大学総合情報処理センターの実システムに組み込み試験運用を通して評価・改善を行う予定である。

参考文献

- [1] Douglas Comer. bit 別冊 第 3 版 TCP/IP によるネットワーク構築 Vol.1., 共立出版
- [2] 大野 尙郎、前田 英明、井田 昌之、松岡 聡、中田 秀基. bit 別冊 Java プログラミング例題集, 共立出版
- [3] 青柳 龍也. Java API プログラミングガイド, クオリティ株式会社
- [4] 有我 成城 他. 一歩先行くインターネット Java 入門, 株式会社翔泳社
- [5] Simple Mail Transfer Protocol RFC821, <http://www.internic.net/rfc/rfc821.txt>
- [6] Post Office Protocol - Version 3, RFC1939, <http://www.internic.net/rfc/rfc1939.txt>