Internetwork Load Testing with Tcl/Tk

Ahmet Can Keskin, Ted Schmitt

Patzschke + Rasp Software AG

Abstract

Networks are the nervous systems of today's economy. It is vital that they work properly and they do not break down under an overload of stress. Detecting and removing the weak points before deployment is a necessity for strategic and economic success. Designing networks for high quality service is a challenging task and automated tests are important for rapid development and optimization of networks. This paper describes a Tcl/Tk based system for comprehensive load testing which alleviates repetitive, time consuming testing tasks.

1 Introduction

With the explosion of the Internet, exponentially increasing traffic, new applications such as multimedia (audio, video streams) etc. network service providers are facing huge demands for high bandwidth and availability to meet Quality of Service (QoS) requirements.

Availability and high throughput are critical success factors for suppliers of large heterogeneous network configurations. This includes internet service providers, enterprise-wide intranets and telecommunication providers. For such organizations designing a network using multiple technologies (ATM, Gigabit Ethernet), new mediums such as xDSL modems is getting more and more a complex task. This involves hardware and software from numerous vendors. It is, therefore, essential that the whole system be tested under real-life load conditions to ensure that all components operate together without error and deliver the required performance to meet QoS issues.

In this paper we present sm@rtTEST the successor of the load testing system presented in [1]. sm@rtTEST is a Tcl/Tk based load test system for entire mid- to large-range enterprise, inter-/intranet and telecommunications networks. For effective and realistic tests sm@rtTEST emulates the behavior of real users, provides means to develop and organize test configurations and monitoring a running test. This system is now currently in use by the Center for Internet and Data Network Platforms, a division of Deutsche Telekom and T-Nova. Mannesmann Arcor will use this system in the near future.

2 Automated Loadtesting

For an end user QoS means quick response times, high availability and reliability. The most realistic test and the best prove that a network will meet its requirements is to stress the network with hundreds or thousands of simultaneous users before deployment. The developers can observe the behavior of the network under load conditions and enhance their understanding of the network system.

An automated testing system emulating hundreds or thousands of virtual users, allowing to organize and drive repeatable tests is the key for effective and realistic loadtesting. Emulating the target number of virtual users firing multiple types of transactions in random combinations is critical for predicting and measuring system behavior under varied conditions. This is one of the main goals of sm@rtTEST.

Further requirements are:

- Scalability regarding the number of automated users.
- Support for a variety of access methods and mediums (PPPoE, xDSL, ATM, analog modem, ISDN modem, S2M).
- Support for a variety of Internet application protocols (HTTP, FTP, Telnet, RTSP, SMTP, POP3) and measurement of quality criteria like performance, response-time, framerate etc.
- Distinct physical connection for each automated user.
- Flexible, easy-to-learn means for defining repeatable test scenarios
- Graphical user interface for organizing, driving, monitoring and analyzing tests

- Extensibility (new application protocols, new access/subscribing methods, analyze tools etc.)

.

3 System Description

The core of sm@rtTEST is the Session Engine providing a runtime-environment for test scripts to emulate real user transactions [Fig. 1]. A concrete load test consists of a specified number of Session Engines running test scripts. A graphical user interface allows the organization, driving and monitoring a running test. As each Session Engine has its own physical connection to the network under test, it gets the entire bandwidth serving for accurate measuring of quality criteria. See the following sections for a detailed description of the system.

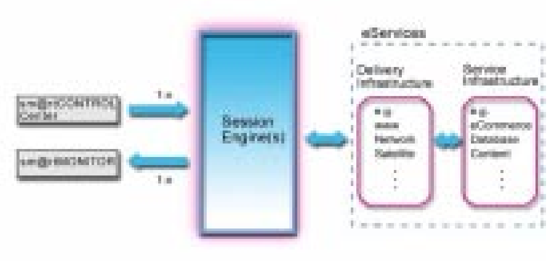

Figure 1: sm@rtTEST Components

3.1 Session Engine

A Session Engine emulates the actions of a human user (e.g. hitting web pages). Every Session Engine can generate a certain load o the system being tested, dependent on the connection (e.g. Modem, ISDN, ADSL). The number of Session Engines used in a testing environment is virtually unlimited and depends only on the hardware available in the test installation.

The Session Engine creates a run-time environment in which a Session Script is run performing a work session. The execution of a Session Script on a Session Engine is the embodiment of an automated user. The Session Engine builds the connection to the system being tested, carries out the transactions specified in the session script and disconnects from the tested system when the session script is complete. The Session Engine can be configured to connect via an analog or ISDN modem, a LAN,/ WAN, and other connection devices.

Many Session Engines may run on a single Session Host. The Session Host is a PC running Linux. The number of Session Engines possible for any Session Host depends on the number of connections possible to the system being tested. One Session Engine for each physical interface (interface port) means that each automated user gets the full bandwidth of the connection device. This is the ideal case since the test environment for the automated user matches that of the real user exactly.

The Session Engine provides also a wide range of access methods: PPP over serial connection, PPP over Ethernet, DHCP. PPP over ATM is currently under development. Also packages for application protocols are provided, these include:

- HTTP, FTP, TELNET, Ping
- SMTP, POP3
- RTSP (Realplayer G2)

3.2 Graphical User Interface

A graphical Control Center [Fig. 2] provides an interactive environment for defining, driving, and coordinating a load scenario. The Control Center is designed using a set of building blocks which allow complete definition and management of a scenario:

- A means to view the test installation and configuration (e.g. session hosts, interface ports)
- A means to define connections to the tested system
- A means to parametrize sessions
- Session management during execution of a load scenario

The Control Center is implemented completely in Tcl/ Tk using the RTL library of VisualGIPSY [7].

The results produced from a run of a load scenario can be viewed graphically with a Monitor [Fig. 3]. The Monitor is a visualization of the measured quality criteria like actual performance, average performance, number of errors etc. The Monitor Component is a Tcl Application built on top of GIPSY, a design tool for the development and prototyping of data visualizations handling dynamic data [6].

3.3 Hardware Configuration

A typical installation of sm@rtTEST consists of one control host and several session hosts. The number of session hosts is dependent on the number of simultaneous connections to be tested.

Each session hosts can support several concurrent Session Engines and various types of network interfaces depending on the desired test configuration. sm@rtTEST supports LAN based Ethernet interfaces connected to ADSL modems as well as analog and digital (ISDN) modems connected via a serial interface.

The number of Session Engines per Session Host depends on the network interface type. With 4-Port Ethernet network adapters and four PCI slots in a PC 16 Ethernet ports allow connection to 16 ADSL modems running 16 Session Engines (and therefore emulation of 16 users) on one Session Host. A S2M-Adapter allows up to 30 ISDN connections, a special RS232 Adapter allows to connect 30 analog or digital modems over a serial connection.

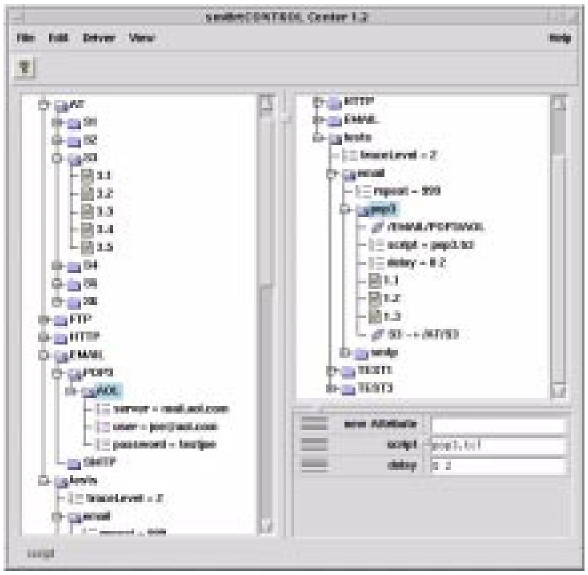

Figure 2: Control Center

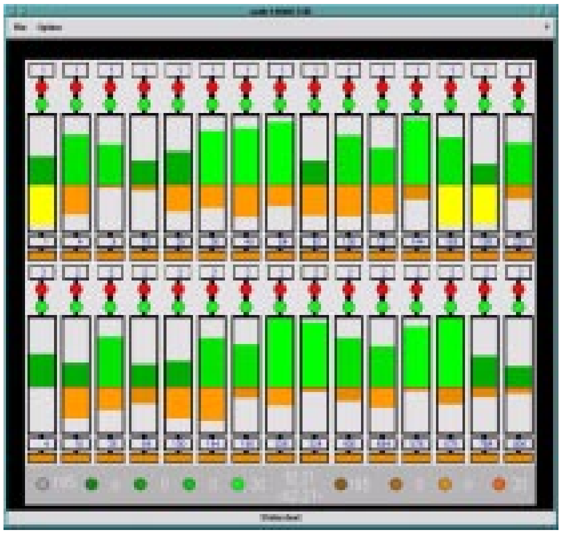

Figure 3: Monitor

4 Implementation

The scripting language Tcl/Tk proved as an ideal platform to develop a full scriptable and extensible load testing system and meet above requirements. The extensibility of Tcl and a large number of available extensions and packages enabled the development of automated users behaving like a real user doing a dial-in, FTP, HTTP, Telnet or downloading a audio/video stream. Due to the gluing capabilities of Tcl it was possible to integrate existent as well as our own networking tools and access clients (e.g. PPP, PPP over Ethernet...).

For an easy and effective session scripting, we implemented session protocol modules handling the client side of FTP, HTTP, Telnet, Ping, SMTP, POP3 and a RTSP Client based on the Realplayer G2 core. Existing packages and extensions are used for the low level protocol handling:

- For FTP we used the FTP-Package of Steffen Traeger and extended it with more flexible time-out mechanisms, a binding to the local network device etc.[3].
- For HTTP we used the standard http package of Tcl.
- With Expect we could integrate existing the standard telnet and ping application of Linux.
- E-mail handling was achieved with ezsmtp [4] and tcllib providing POP3 handling functions.
- With the C/C++ based Realplayer SDK we implemented a Tcl extension for the integration in sm@rtTEST.

Session Scripts are written completely in Tcl using the above protocol modules. The Session Engine runs such a script in a Tcl slave interpreter.

The Control Center displays a load test configuration as a tree-structure for an easy creation, organization and management of load tests. The manipulation of load test is fully GUI based with Drag&Drop functionality. For the tree representation we used the tree widget of the RTL library of VisualGIPSY [7]. The RTL library is a collection of complex widgets like tree widget, tab widget etc.

5 Future Directions and Conclusion

For further development of sm@rtTEST there are a lot of possibilities:

- Increase the number of Session Engines on one Session Host to 1000-2000 Sessions. As the physical

ports will hardly reach such a high number, the Session Engines have to share the physical ports and the available bandwidth.
- New access protocols and methods like L2TP, AODI, ATM
- Further application protocols like MS Media, Voice over IP.
- Flexible online and offline analyze tools

Today's IT-Systems include a mix of legacy applications, distributed client/server, Enterprise Resource Planing (ERP) and transaction based web applications, woven together with sophisticated networking systems. With the emerging complexity of such systems the requirements on comprehensive load testing will grow. sm@rtTEST with its scripting capabilities and its open architecture provides a flexible platform to meet future requirements on comprehensive loadtesting.

References

[1] Ahmet Can Keskin, Till Immanuel Patzschke, Ernst von Voigt. *"Tcl/Tk: A Strong Basis for Complex Load Testing Systems"*. Proceedings of the 7th USENIX Tcl/Tk Conference. February, 2000.

[2] sm@rtTEST: The loadtest system described in this paper is marketed by interNetwork AG. More information may be obtained from their web site under http://www.internetwork-ag.de

[3] FTP Library Package http://home.t-online.de/home/Steffen.Traeger/tindexe.htm

[4] D. J. Hagberg: Easy SMTP. http://www.millibits.com/djh/tcl

[5] Libes, D., *"Using Expect to Automate System Administration Tasks"*, Proceedings of the 1990 USENIX Large Systems Administration Conference (LISA) IV, Colorado Springs, CO, October 17-19, 1990

[6] GIPSY is a Tcl/tk-based platform-independent visualization software which is also provided from PRS. Information on the web: http://www.prs.de/int/products/gipsy

[7] VisualGIPSY is Tcl/Tk GUI builder developed by PRS. Information on the web: http://www.prs.de/int/products/vg/index.html

[8] Newport Group Inc., *"Load Testing Gains Steam as a Mainline Business Practice"*, 1999. http://www.newport-group-inc.com

Address of the Authors:

Patzschke + Rasp Software AG
Bierstadter Straße 7
D-65189 Wiesbaden

Germany

www.prs.de - info@prs.de