WIDE Technical-Report in 2008

Gulliver Project : Building Distributed Active Measurement Appliances

wide-tr-mawi-gulliver-00.pdf

WIDE Project : http://www.wide.ad.jp/

If you have any comments on this document, please contact to ad@wide.ad.jp

Gulliver Project : Building Distributed Active Measurement Appliances

Yuji Sekiya, Yohei Kuga and Kenjiro Cho

January 4, 2008

1 Overview

This paper describes the design and implementation of a plug-and-measure appliance and its remote management framework. We use a small diskless box as an active measurement appliance, and the remote management framework is built into the firmware of the appliance. A management server, by both a GUI or scripts, remote appliances can be instructed to preform measurements, current appliance status, and apply firmware updates. As a case study of this system , we show long-term DNS measurements obtained from the appliances deployed around the world.

The purpose of this project is to build a measurement appliance and framework which can easily be deployed in a developing world instructor. The core goals of the framework are (1) distributing probe boxes all over the world, (2) managing the boxes easily, and (3) getting measured results easily. We call out framework the "Gulliver Framework". In this section the overview of the appliance and the framework is described.

In order to build a distributed active measurement framework, there are several requirements to achieve. The requirements for the framework are (1) easy installation and robustness of probe boxes, (2) security of the probe box and communication, (3) low costs for management, and (4) applicability of various active measurements.

In consideration of the above requirements, we have selected to use the SEIL appliance, rather then a PC as a probe box. SEIL is a commercial product produced by the Internet Initiative Japan (IIJ) Inc, developed for use as a SOHO router. The advantages of using this platform for us are: (1) the hardware has been proven to work for a 24x7 operation, (2) a remote management system for a SOHO router use is already developed, and (3) we can easily import future bug or security fixes from the commercial version of the product. We have customized the SEIL's firmware in order to support our measurement activies. Should the need arise we could easily port the framework to some other hardware.

Most of the existing measurement efforts use PCs as measurement boxes. In constast to a PC an applications: is cheaper to purchase, cheaper to send, take up less space at the hosting institution, and is has a lower power consumption. In addition SEIL's disk-less storage means improved reliablity in the face of unreliable power supplies, common some developing regions. makes it a more superior choice given our target hosting sights.

As for security, the communication between the probe boxes and the servers should be encrypted. The probe box should accept only connections from our specified servers. Even in the unlikely case that one one of the box is physically stolen, the over all security of the framework should not be threatened.

The probe boxes should be managed and operated in an integrated fashion by the management server. One of the goals of the project is to deploy more than 100 probe boxes world wide. If a vulnerability is found in the firmware or if we want to build new measurement tools into the firmware, it should be upgraded in all the probe boxes without pains and troubles. In addition to the firmware upgrading, daily monitoring of operations should be provided.

The firmware of our appliances is a derivative of NetBSD, so measurement tools which run on NetBSD can be easily be deployed in future firmware updates. This provides an easy mechanism for development and debugging for measurement tools. The firmware also includes a remote management system. All the appliances can be controlled and monitored from the management server through the remote management system, called SEIL Management Framework(SMF).

2 Framework Design

The framework consists of probe appliances, management servers and measurement collection servers. We have designed the framework based on the client-server model. The set of servers manage, control and monitor all probe appliances. The appliance does not start any measurement action until it is instructed to do so by the management server.

2.1 Architecture Goals

As described before, the appliance has the remote management system built into the firmware. The Gulliver Framework has the following architectural goals.

(a) plug-and-measure of appliances; even if the network environment which the appliance is located in is changed and an administrator can not login the appliance remotely, the configuration of the appliance can be changed at the management server and just cycling power of the appliance, the appliance will boot with new applicable configuration. We want to distribute appliances without circumstance in developing countries.

(b) low operational costs and effective measurements; when the number of activated appliances is increased, the operational costs is not increased proportionately. An administrator can control a number of appliances as controlling just an appliance. However, the measured results are increased proportionately and an administrator can collect the results easily.

(c) secure and safe against snooping and cracking; the appliance talks to the servers with authenticated and encrypted datagrams, and the measured results also should be retrieved securely.

2.2 Appliance bootstrap

When the appliance is connected to a UTP cable and power source, it obtains an IP address using DHCP, and then the appliance bootstraps as shown in figure 1. The appliance computes its own unique ID from its MAC address. The appliance knows only the IP address of a master server and the server's SSL certificate.

The master server's only job is to authenticate the appliance from its ID and tell the appliance about a second-level server. At bootstrap, the appliance contacts the master server using XML-RPC over SSL, verifies the server using the built-in certificate, and passes its unique ID to the server. The master server checks the appliance's ID, and returns the IP address and the SSL certificate of a second-level server.

The appliance contacts the second-level server using XML-RPC over SSL. The reasoning behind this server hierarchy is to allow a set of second-level servers for future scalability as well as to have completely different services for the appliances. The second-level server returns the appropriate configuration for the appliance, if the authentication is succeeded. The configuration includes the SSL certificates of the servers, and the SSH public key of a measurement collection server. The appliance keeps this configuration on memory, and reboots itself using this configuration.

2.3 Appliance management

When controlling the appliances after a reboot, firmware upgrading or requesting status reports, an administrator does not need to send the requests directly to every appliance. Rather, all he or she needs to do is send the requests to the second-level server as shown in figure 2.

For example, if an administrator wants to reboot Box B and C, she can send a reboot request to the second-level server with the IDs of Box B and C, then the second-level server sends reboot requests to Box B and C.

When an appliance talks to a server, it uses XML-RPC over SSL. On the other hand, when a server talks to an appliance, it sends a command

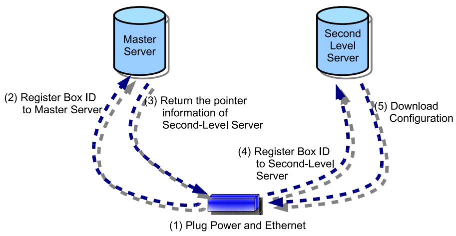

Figure 1: The flow diagram of bootstrap

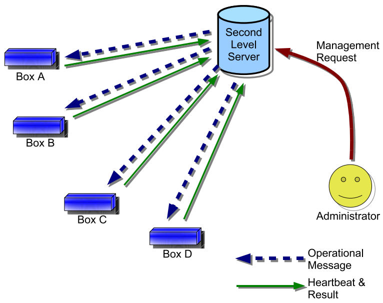

Figure 2: The flow diagram of management

over SSH. In addition to these communications, appliances can periodically send heartbeats to a server; a heartbeat message includes the status of the interfaces, CPU and memory so that administrators can monitor the status of the deployed appliances.

2.4 Appliance measurement

Administrators can send start and stop requests for measurement programs to the appliances in a similar manner as the appliance management shown in figure 3. When an appliance receives a start request from second-level server, it starts the measurement. When the measurement is completed, the appliance sends the results to the measurement collection server specified by the administrator.

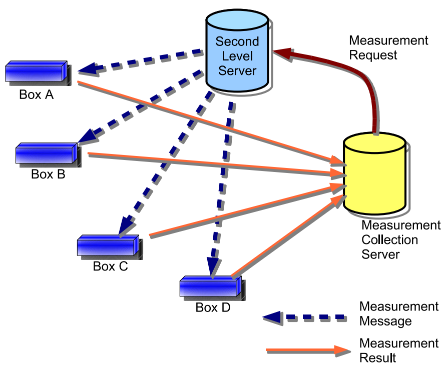

Figure 3: The flow diagram of measurement

3 Framework implementation

There are a number of challenges in building the appliances and framework. In this section, we have described details of our implementation, especially about our modifications to the original SMF and challenges encountered.

The first challenege we faced was to reduce firmware memory usage. The appliance has only 64MB memory. In order to save memory we deleted unused functions for active measurements from the original firmware.

The second challenge was building a wrapper software to run the existing measurement tools without modifications. The wrapper was integrated into the management system firmware.

The third challenge was handling static configuration of IPv4 addresses. The original remote management system, SMF only supports DHCP environments. This means that if a DHCP server is not available, the appliance can not get IPv4 address or communicate with the master server. Depending on the policy of the instillation site this can be problematic, because there exist situations were only a static IPv4 address is available. In order to adapt to a static IPv4 address, we modified the firmware so that the IPv4 address and default gateways could be pre-configured. If the IPv4 address is pre-configured, the appliance communicate to the master server using the address, then it boots in a manner similar to a DHCP environment.

The fourth challenge was the to build a more user-friendly administrative interface. Thus reducing the cost to manage and monitor the appliances. It is not user-friendly to send messages using XML-RPC with SSL for users who would like to just run active measurements on this framework, so it is very important to deploy and generalize the framework for various active measurements. The interface is shown in figure 4.

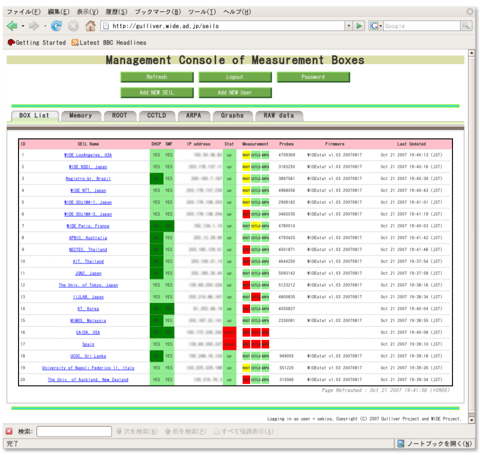

Figure 4: Web GUI for the managements

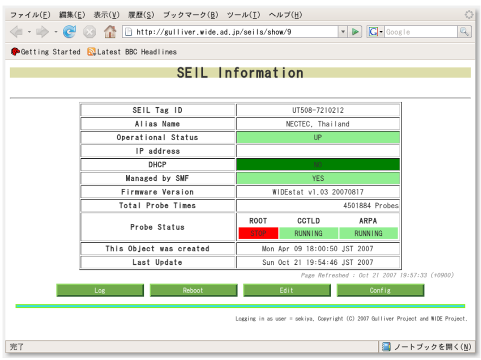

Figure 5: Details of the appliance

In the figure there is a list of the registered appliances. By clicking the name of appliance on the list, the detail of the appliance is shown as in figure 5. By clicking “reboot” button in the figure, the reboot request will be sent to the appliance through the second-level server and the appliance will be rebooted.

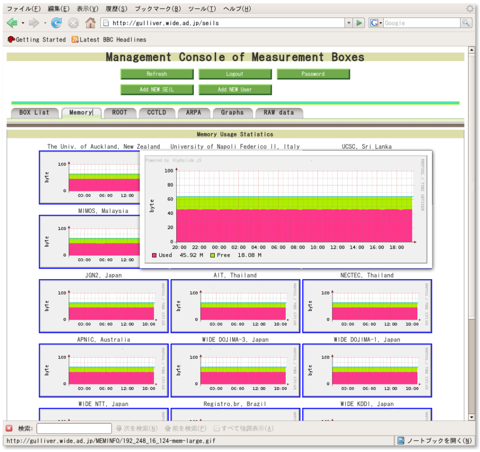

Figure 6: Monitoring memory usages

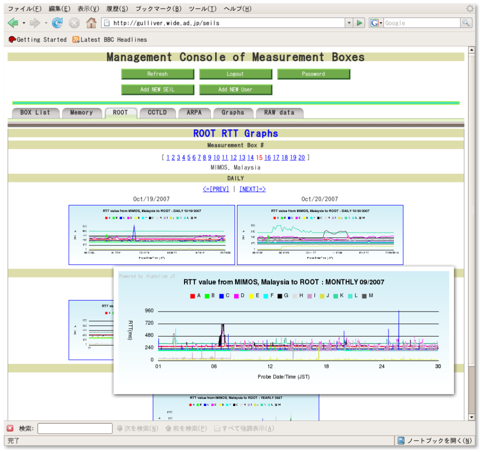

Figure 7: Monitoring measurement status

Figure 6 shows the memory usages of each appliance. The status is monitored by heartbeats which every appliance send to the second-level server. Figure 7 also shows the status of the measured results collected by measurement collection server. As shown in the figures, administrators can manage and monitor the appliances using web GUI.

4 Measurements

As of October 2007, 18 appliances are distributed and activated. Table 1 summarizes the locations. Only one box is located in United States, while the majority are located in Asian countries reflecting the project aims to deploy the appliances in developing regions.

Two measurement tools are implemented in the appliance at this time, dnsprobe and scamper. As a case study of the Gulliver framework, we run two measurements and validate and prove advantages of the Gulliver Framework. The results are described in this section.

5 Project Status

A further directions of this study are (1) distributing more appliances all over the world at least 100 boxes, especially developing regions, (2) implementing

Table 1: Location of measurement boxes

Box ID	Organization	City	Country
1	WIDE	Los Angenes	U.S.A.
2	WIDE	Tokyo	Japan
3	Registro.br	Sao Paulo	Brazil
4	WIDE	Tokyo	Japan
5	WIDE	Osaka	Japan
6	WIDE	Osaka	Japan
7	WIDE	Paris	France
8	APNIC	Brisbane	Australia
9	NECTEC	Bangkok	Thailand
10	AIT	Pathumthani	Thailand
11	JGN2	Tokyo	Japan
12	The University of Tokyo	Tokyo	Japan
13	IIJ Inc.	Tokyo	Japan
14	Korea Telecom	Seoul	Korea
15	MIMOS	Kuala Lumpur	Malaysia
16	CAIDA	San Diego	U.S.A.
17	University of Colombo School of Computing	Colombo	Sri Lanka
18	University of Napoli Federico II	Napoli	Italy
19	The University of Auckland	Auckland	New Zealand

module mechanism in the firmware in order to load and unload measurement tools dynamically, and (3) enabling managements even if the appliance is located behind NAT.

The status of gulliver project is available on http://gulliver.wide.ad.jp/

Copyright Notice