

GENI

Global Environment for Network Innovations

Milestone S2.i Implement and Integrate Measurement Handler

Document ID: GENI-MS2I-LEARN-Mar16

March 16, 2010

Prepared by:

D. Majumder¹, M. S. Wang², I. Baldine³, D. Gurkan¹, C. P. Lai², K. Bergman²

1: University of Houston: College of Technology

2: Columbia University: Dept. of Electrical Engineering

3. RENCi, UNC-Chapel Hill

Under Project Nr. 1733

“Programmable Measurements over Texas-based Research Network: LEARN”

TABLE OF CONTENTS

1	DOCUMENT SCOPE.....	3
1.1	EXECUTIVE SUMMARY	3
1.2	RELATED DOCUMENTS	3
	GENI Documents	3
1.3	DOCUMENT REVISION HISTORY	3
2	PREVIOUS WORK	4
3	IMF ARCHITECTURE OVERVIEW.....	4
4	MEASUREMENT HANDLER SOFTWARE OVERVIEW.....	6
4.1	TL1 TOOLKIT	6
4.2	SUPPORTING MODULES.....	6
4.3	SERVER PROGRAM.....	6
	4.3.1 Login Procedure	7
	4.3.2 Measurement Retrieval Procedure.....	7
	4.3.3 Logout Procedure	8
	4.3.4 Remarks	8
5	SUMMARY AND CONCLUSIONS	9
6	BIBLIOGRAPHY	9

1 Document Scope

This section describes this document's purpose, its context within the overall GENI project, the set of related documents, and this document's revision history.

1.1 Executive Summary

This technical note presents the results obtained in work package "Milestone S2.i: Implement and Integrate Measurement Handler" of Project Nr. 1733, "Programmable Measurements over Texas-based Research Network: LEARN" [LEARN_1].

This document is a report on the implementation of the Measurement Handler Software (MHS) and its integration with the Integrated Measurement Framework (IMF). MHS is a set of Perl scripts that enables one to obtain optical measurements (BER and optical power) from a variety of performance monitoring devices, including the Infinera Digital Transport Node (DTN) and the Polatis switch. Further, IMF is a joint project among research teams from the University of Houston, University of North Carolina at Chapel Hill/RENCI, Columbia University, and North Carolina State University. At University of Houston (GENI-LEARN), our task is to integrate the MHS with the IMF such that we can obtain BER and optical power measurements using the 4 DTNs located in the ORCA-BEN network.

In Section 2, we summarize the work of our previous spiral 2 milestones. In Section 3, we present an overview of IMF and list our contributions in its implementation. In Section 4, we describe the software implementation of the MHS. Section 5 gives a summary and conclusion.

1.2 Related Documents

The following documents are related to this document, and provide background information, requirements, etc., that are important for this document.

GENI Documents

Document ID	Document Title and Issue Date
GENI-S2H-LEARN-Nov09	Draft Measurement Data File Format, November 16, 2009
ERM_S2a_Dec09	Unified Measurement Framework: NetFPGA Cube Prototype, December 03, 2009
ERM_S2b_Mar10	Demonstrating Embedded Real-Time Physical Measurement from ORCA-BEN Substrate, March 04, 2010
GENI_MS2H_part2_LEARN_Jan08_msw	Specifications for the Measurement Handler Software, January 08, 2010
IMF Architecture	Milestone S2.d, IMF Project

1.3 Document Revision History

The following table provides the revision history for this document, summarizing the date at which it was revised, who revised it, and a brief summary of the changes. This list is maintained in chronological order so the earliest version comes first in the list.

Revision	Date	Revised By	Summary of Changes
1.0	03/08/10	D. Majumder	Initial draft
1.0	03/08/10	C. Lai	Revised Update
1.0	03/09/10	D. Gurkan	Revised Update
1.0	03/10/10	M. Wang	Revised Update

2 Previous Work

The milestone on the drafting of the data file format for the transfer of measurement data between the Infinera Digital Transport Node (DTN) [Infinera_1, Infinera_2] and the Integrated Measurement Framework [IMF_1] has been submitted during GEC6, November 16, 2009 [LEARN_2]. We explained how the measurement data file format is based on the TL1 command format of the Infinera DTN [Infinera_3, Infinera_4]. Also in [LEARN_2], we examined two specific TL1 commands that are of particular interest for the GENI measurement framework: an optical power measurement and the bit-error rate (BER) measurement. On January 8, 2010, a report on the specifications for the Measurement Handler Software was submitted.

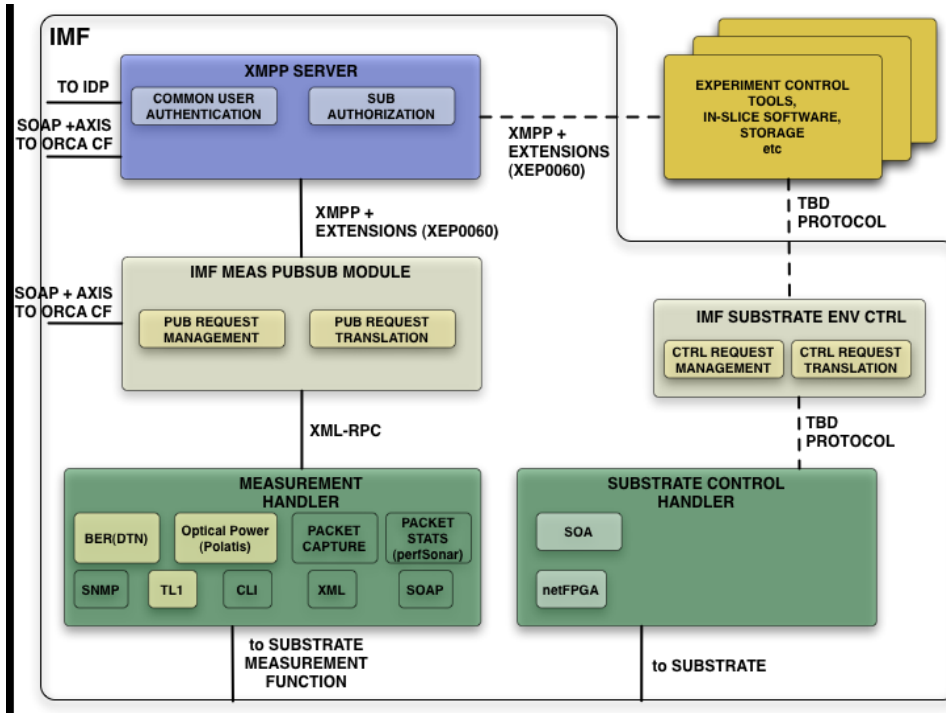
3 IMF Architecture Overview

The below figure shows the components needed to implement the IMF. A more detailed version of the IMF architecture description is presented in [imf_1]. In this section, we describe the component architecture in the context of LEARN's contribution.

IMF is implemented as a set of federated servers implementing Publish/Subscribe or *PubSub* functionality on an XMPP server [xmpp_1], [xmpp_2], [xmpp_3], [xmpp_4]. Using the PubSub module, consumers of measurement data would subscribe to a particular measurement data of interest and ignore the unwanted measurement data. Once the underlying substrate has obtained the result, the data is published and only the subscribers of that data will receive the result.

XMPP Server

- Track subscription requests from measurement data consumers.
- Receive publish events from the measurement PubSub Module, which contain measurement data or meta-data.
- LEARN: we setup and tested the correct functionality of this server within ORCA-BEN.



Measurement PubSub Module (PSM)

- Allows consumer to subscribe only to the measurement data one is interested in and receives the published data at the frequency that the data is generated, thus removing the need for explicit polling by data consumers.
- Translate information between physical topology of the substrate and virtual topology of the slice.
- LEARN: we setup and tested the correct functionality of this module as it receives subscribe requests of measurement data consumers and receives published results from the underlying substrate.

Measurement Handler (MH)

- Presents a uniform interface to configure and query substrate measurement capabilities.
- LEARN: we developed MH for the Infinera DTN to enable it to measure BER and optical power.

Substrate Environment Control module (SEC)

- Same function as PSM module, but applied to substrate environment control, rather than measurement data collection.
- ERM goal: This is a newly defined functionality that will be implemented in the future.

Substrate Control Handler (SCH)

- Presents a uniform interface to functions and components within the substrate that can help control and manipulate the experiment environment.

- ERM goal: This is a newly defined functionality that will be implemented in the future.

4 Measurement Handler Software Overview

The Measurement Handler Software (MHS) is a server program, which communicates with Integrated Measurement Framework (IMF) using the XML-RPC protocol. MHS supports invocation of 3 procedures/functions:

- a. connectToDevice – logs into the device (Infinera/Polatis),
- b. retrieveMeasurement – fetch measurement from device,
- c. disconnectFromDevice – logs out from device.

It supports multiple login and simultaneous retrieval of measurements. The following measurement types are currently supported:

- Infinera: PREFEC-BER, POSTFEC-BER and PORTPOWER (received power),
- Polatis: PORTPOWER.

4.1 TL1 Toolkit

TL1 Toolkit [SARA_1] has been used for the general access and parsing functions for TL1 interface with Infinera and Polatis devices. However, the following modifications were necessary to ensure multi-user access:

- a. Unique connection identifier for each login (tl1 object),
- b. Thread-safe response fetch for concurrent requests,
- c. Separate termination of response retrieval for each logout,
- d. Fetching of autonomous messages (response without any requests, like error message, etc) from the device for debugging purpose.

4.2 Supporting Modules

The extraction of specific measurements from the raw response message from the device is done in separate perl modules:

- pol_tl1.pm
- infinera_tl1.pm

Based on the expected format of response, the particular data field is returned. In the case of an abnormal response format, an error message is returned.

4.3 Server Program

XMLRPC_API_Server_MH_InfineraPolatis.pl is the main server program which creates a daemon process that continuously polls a port (currently 8001) for XML-RPC requests. It accepts requests to connect, disconnect, and perform measurements as described earlier.

4.3.1 Login Procedure

For a connectToDevice call, the request should have the format:

```
<?xml version="1.0" encoding="UTF-8"?> <methodCall>
<methodName>connectToDevice</methodName>
<params>parameters</params>
</methodCall>
```

where the parameters/arguments are:

- deviceIP=<IP address or FQDN of device>
- port=9090/3082
- username=<username>
- password=<pwd>
- device=Infinera/Polatis (case-insensitive)

Using these parameters, a new t11 connection is created with the device and a unique connection identifier is returned to the client in the XML-RPC structure:

```
<?xml version="1.0" encoding="UTF-8"?><methodResponse><params><param><value><struct>
<member><name>connectionObject</name><value><int>UNIQUE ID</int></value></member>
</struct></value></param></params></methodResponse>
```

In case of login failure, the appropriate error message is returned. The procedure can accept concurrent requests. Every connection is automatically logged off if it is idle for more than 5 minutes.

4.3.2 Measurement Retrieval Procedure

For a retrieveMeasurement call, the request should have the format:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall><methodName>retrieveMeasurement</methodName>
<params>PARAMETERS</params>
</methodCall>
```

where the following parameters are expected:

- connectionObject=ID(Id returned by the connectToDevice method)
- chassis=<val> DLMslot=<val> opticalChannel=<val> (for Infinera)
- port=<val> (for Polatis)
- measureType=prefec-ber/postfec-ber/portpower(for Infinera) portPower(for Polatis)

The measureType is case-insensitive.

In the case of successful retrieval, the result is returned in the format:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse><params><param><value><struct><member>
<name>MEASURETYPE</name><value><double>MEASUREMENT_VALUE</double></value>
</member></struct></value></param></params></methodResponse>
```

The function can fetch concurrent requests from the same as well as different connection ids. In the case of failure, an appropriate error message is returned with the <name> field as ERROR instead of MEASURETYPE.

4.3.3 Logout Procedure

For a disconnectToDevice call, the request should have the format:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall><methodName>disconnectFromDevice</methodName>
<params>PARAMETERS</params></methodCall>
```

where the parameters are:

- connectionObject=ID (Id returned by the connectToDevice method)
- deviceIP=<IP address or FQDN of device>
- port=9090/3082
- username=<username>
- password=<pwd>
- device=Infinera/Polatis

The username and password is required to prevent erroneous disconnections. For successful disconnection, the response will be sent in the below format:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse><params><param><value><struct><member>
<name>Disconnected</name><value><int>ID</int></value>
</member></struct></value></param></params></methodResponse>
```

Concurrent disconnection requests can be processed.

4.3.4 Remarks

For any of the above requests, if the device (Infinera/Polatis) does not respond within 200 seconds (3min 20 sec), the request is canceled and an error message is returned.

The XMLRPC_API_Server_MH_InfineraPolatis.pl requires the below files to be placed in the same folder:

- tl1.pm
- pol_tl1.pm
- infinera_tl1.pm

It also requires the below perl modules to be installed:

- threads;
- threads::shared;
- Thread::Queue::Any;
- Net::Telnet;
- Date::Manip;
- CGI::Cookie;
- SOAP::Lite;
- Thread::Semaphore;

The program is to be run as:

```
perl XMLRPC_API_Server_MH_InfineraPolatis.pl
```

It prints the login, logout, and error details with the corresponding time-stamps.

Screenshot of sample output:

Contact to XMLRPC server at <http://geni-imf-dev:8001>

2010-03-09 19:11:48: ID1001-Logged into dtn-1.duke.ben Infinera.

2010-03-09 19:11:48: ID1002-Logged into os.duke.ben Polatis.

2010-03-09 19:29:53: ID1001-Logged out from dtn-1.duke.ben Infinera.

2010-03-09 19:29:53: ID1002-Logged out from os.duke.ben Polatis.

5 Summary and Conclusions

In this report, we drafted the specifications for integrating MHS with IMF. We previously defined the measurement data file format as the TL1 response format of the Infinera DTN and the specifications for the measurement handler software.

6 Bibliography

- [1] [LEARN_2] D. Gurkan, Y. Xin, M. S. Wang, C. P. Lai, and K. Bergman, "Specifications for the Measurement Handler Software" [Online]. Available: http://groups.geni.net/geni/attachment/ticket/270/GENI_MS2H_part2_LEARN_Jan08.pdf
- [2] [DMEAS_2] Data Plane Measurements: DMEAS [Online]. Available: <http://groups.geni.net/geni/wiki/Data%20Plane%20Measurements>
- [3] [LEARN_1] Programmable Measurements over Texas-based Research Network: LEARN [Online]. Available: <http://groups.geni.net/geni/wiki/LEARN>
- [4] [ERM_1] C. P. Lai, M. S. Wang, K. Bergman, "Unified Measurement Framework: NetFPGA Cube Prototype," Dec. 2009 [Online]. Available: http://groups.geni.net/geni/attachment/ticket/279/ERM_S2a_Dec09.pdf
- [5] [ERM_2] M. S. Wang, C. P. Lai, and K. Bergman, "Demonstrating Embedded Real-Time Physical Measurement from ORCA-BEN Substrate", Mar. 2010 [Online]. Available: http://groups.geni.net/geni/attachment/wiki/Embedded%20RealTime%20Measurements/ERM_S2b_Mar10.pdf?format=raw
- [6] [imf_1] I. Baldine, K. Bergman, R. Dutta, D. Gurkan, C. Lai, G. Rouskas, A. Wang, M.S. Wang, "IMF Architecture (Project Nr. 1718, Milestone S2.d)," (2010, Feb) [Online]. Available: <https://geni-imf.renci.org/trac/wiki>
- [7] [Infinera_1] "Infinera DTN Hardware Description Guide," Release 5.0, Version 005, Document ID 1900-215.
- [8] [Infinera_2] Infinera DTN [Online]. Available: <http://www.infinera.com/products/DTN.html>
- [9] [LEARN_2] M. Wang, D. Gurkan, C. P. Lai, K. Bergman, "Draft Measurement Data File Format," November 2009 [online]. Available: http://groups.geni.net/geni/attachment/ticket/270/GENI_S2H_LEARN_Nov09.pdf
- [10] [SARA_1] R. van der Pol, A. Toonk, "Data Gathering in Optical Networks with the TL1 Toolkit," Passive and Active Measurements Conference (PAM), 2009.
- [11] [xmpp_1] RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core
- [12] [xmpp_2] RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence
- [13] [xmpp_3] XEP-0060: Publish Subscribe (<http://xmpp.org/extensions/xep-0060.html>)
- [14] [xmpp_4] OpenFire XMPP server (<http://www.igniterealtime.org/projects/openfire/>)