

CS 461 - Fall 2016 - Design Document

Matthew Johnson, Cody Malick, Garrett Smith
Team 51, Cloud Orchestra

Abstract

This document outlines the design considerations for the implementation of Open vSwitch and other networking technologies in the Cloud Integrated Advanced Orchestrator (Ciao). It describes the various techniques, structure, and technology choices that will be used in the execution of our project.

CONTENTS

I	Introduction	2
II	Overview	2
II-A	Scope	2
II-B	Purpose	2
II-C	Intended audience	2
II-D	Conformance	2
III	Definitions	2
IV	Conceptual model for software design descriptions	2
IV-A	Software design in context	2
IV-B	Software design descriptions within the live cycle	2
IV-B1	Influences on SDD preparation	2
IV-B2	Influences on software life cycle products	2
IV-B3	Design verification and design role in validation	2
V	Design description information content	3
V-A	Introduction	3
V-B	SDD identification	3
V-C	Design stakeholders and their concerns	3
V-D	Design views	3
V-E	Design viewpoints	3
V-F	Design elements	3
V-F1	Design entities	3
V-F2	Design attributes	3
V-F3	Design relationships	3
V-F4	Design constraints	3
V-G	Design overlays	3
V-H	Design rationale	3
V-I	Design languages	3
VI	Design viewpoints	3
VI-A	Introduction	3
VI-B	Context viewpoint	3
VI-B1	Design concerns	3
VI-B2	Design elements	3
VI-B3	Example languages	3
VII	High-level considerations	3
VIII	Conclusion	3
	References	3

I. INTRODUCTION

Our project is to first switch the Linux-created GRE tunnel implementation in Ciao to use GRE tunnels created by Open vSwitch. From that point we will switch the actual tunneling implementation from GRE to VxLAN/nvGRE based on performance measurements of each on data center networking cards. After this is completed, a stretch goal is to replace Linux bridges with Open vSwitch switch instances. This document outlines the steps, techniques, and methodology we will utilize to achieve each goal.

II. OVERVIEW

- A. *Scope*
- B. *Purpose*
- C. *Intended audience*
- D. *Conformance*

III. DEFINITIONS

Bridge	Software or hardware that connects two or more network segments.
Ciao	Ciao is a cloud orchestrator that provides an easy to deploy, secure, scalable cloud orchestration system which handles virtual machines, containers, and bare metal apps agnostically as generic workloads. Implemented in the Go language, it separates logic into "controller", "scheduler" and "launcher" components which communicate over the "Simple and Secure Node Transfer Protocol (SSNTP)" [1].
Cloud	A huge, amorphous network of servers somewhere [2].
Cloud Orchestration	A networking tool designed to aid in the deployment of multiple virtual machines, containers, or bare-metal applications [1].
CNCI	Virtual Machines automatically configured by the ciao-controller, scheduled by the ciao-scheduler on a need basis, when tenant workloads are created [3].
Generic Routing Encapsulation (GRE)	Encapsulation of an arbitrary network layer protocol so it can be sent over another arbitrary network layer protocol [4].
Linux Bridge	Configurable software bridge built into the Linux kernel [5].
Network Node (NN)	A Network Node is used to aggregate network traffic for all tenants while still keeping individual tenant traffic isolated from all other the tenants using special virtual machines called Compute Node Concentrators (CNCIs) [3].
nvGRE	Network Virtualization using Generic Routing Encapsulation [6].
Open vSwitch	Open source multilayer software switch with support for distribution across multiple physical devices [7].
OVS	Open vSwitch [7].
Packet Acceleration	Increasing the speed of the processing and transfer of network packets.
Packet Encapsulation	Attaching the headers for a network protocol to a packet so it can be transmitted using that protocol [8].
SSNTP	The Simple and Secure Node Transfer Protocol (SSNTP) is a custom, fully asynchronous and TLS based application layer protocol. All Ciao components communicate with each others over SSNTP [9].
Tunnel	Point to point network connection that encapsulates traffic between points [8].
VxLAN	Virtual Extensible Local Area Network [10].

IV. CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS

- A. *Software design in context*
- B. *Software design descriptions within the live cycle*
 - 1) *Influences on SDD preparation:*
 - 2) *Influences on software life cycle products:*
 - 3) *Design verification and design role in validation:*

V. DESIGN DESCRIPTION INFORMATION CONTENT

- A. *Introduction*
- B. *SDD identification*
- C. *Design stakeholders and their concerns*
- D. *Design views*
- E. *Design viewpoints*
- F. *Design elements*
 - 1) *Design entities:*
 - 2) *Design attributes:* Name attribute
Type attribute
Purpose attribute
Author attribute
 - 3) *Design relationships:*
 - 4) *Design constraints:*
- G. *Design overlays*
- H. *Design rationale*
- I. *Design languages*

VI. DESIGN VIEWPOINTS

- A. *Introduction*
- B. *Context viewpoint*
 - 1) *Design concerns:*
 - 2) *Design elements:*
 - 3) *Example languages:*

VII. HIGH-LEVEL CONSIDERATIONS

Our software defined network will be written in the Go programming language and fully integrated in to the Cloud Integrated Advanced Orchestrator (Ciao) [1]. The Go programming language was selected for several reasons, including the efficiency of the language regarding both speed and memory, the concurrency capabilities, and the ease of implementation. Go was compared against C and Python as alternatives, and prevailed in every criteria except for availability of the language. This network mode will be written as a standalone networking mode for Ciao as an additional option to the standard Linux bridges available now. For this reason it must be fully integrated with the Ciao networking framework as it currently exists [3].

VIII. CONCLUSION

We have outlined the steps and design strategy we will take for each goal. Our design methodology is incremental design, starting with the first goal (Open vSwitch-created GRE tunnels) and incrementing through each feature until all goals are achieved.

REFERENCES

- [1] T. Pepper, S. Ortiz, M. Ryan *et al.* (2016, sep) Ciao readme. [Online]. Available: <https://github.com/01org/ciao/blob/master/README.md>
- [2] R. Munroe. (2011, jun) The cloud. [Online]. Available: <http://xkcd.com/908/>
- [3] M. Castelino. (2016, may) Ciao networking. [Online]. Available: <https://github.com/01org/ciao/blob/master/networking/README.md>
- [4] F. T. Hanks, Li. (1994, oct) Generic routing encapsulation (gre). [Online]. Available: <https://tools.ietf.org/html/rfc1701>
- [5] T. L. Foundation. (2016, nov) Bridge. [Online]. Available: <https://wiki.linuxfoundation.org/networking/bridge>
- [6] M. Sridharan, A. Greenberg, N. Venkataramiah, K. Dudam, I. Ganga, and G. Lin. (2015, sep) Rfc7637. [Online]. Available: <https://tools.ietf.org/html/rfc7637>
- [7] (2016, nov) Open vswitch. [Online]. Available: <http://www.openvswitch.org/>
- [8] J. Kurose and K. Ross, *Computer Networking*, 6th ed. Pearson, 2012.
- [9] S. Ortiz, J. Andersen, and D. Lespiau. (2016, sep) Simple and secure node transfer protocol. [Online]. Available: <https://github.com/01org/ciao/blob/master/ssntp/README.md>
- [10] M. Mahalingam. (2014, aug) Virtual extensible local area network (vxlan): A framework for overlaying virtualized layer 2 networks over layer 3 networks. [Online]. Available: <https://tools.ietf.org/html/rfc7348>