**Nick Karanatsios**

Manager

1st Software Development Department

IP Business Division

1132, Hinode, Abiko, Chiba, 270-1198, Japan

phone:+81-4-7185-6912

email:k-nick@pb.jp.nec.com

**Interests**

Ad-Hoc pervasive networks, software defined networking; web services frameworks for distributed enterprise applications, and generic algorithms for various computational tasks.

**Experience**

NEC Engineering, Ltd. Abiko, Chiba, Japan

Software engineer (Manager) April 2002 - Present

**Research prototype development for security information event management system (2016/11 ~ present)**

Collect security information logs normalize, enrich and filter for analysis preparation.

Create JsonLogic rules to identify alerts or potential threats from the logs in real-time.

**Development of JsonLogic rule engine compiler prototype (2016/8 ~ 2016/9)**

Design and coding of <http://jsonlogic.com/> rule engine in ruby for processing queries to extract and transform data security information.

**Development of cyber physical security research prototype (2016/6 ~ 2016/10)**

Design and coding of Policy Decision Controller of the cyber physical prototype. Policy Decision Controller directs requests received from Policy Administrator Controller to pertaining security subsystems.

**Development of cyber physical security research prototype (2016/6/13 ~ 2016/6/23)**

Draft design specifications of cyber physical security prototype interfacing to dynamically configurable security subsystems.

**Cyber-physical security specifications (2016/4~2016/5)**

Draft the specifications for the second development phase of the cyber-security subsystem.

The main objective was to make the policy administration subsystem agnostic of the security subsystems.

**Information security development phase 1(2016/4~2016/6/28)**

Rapid prototype development for a framework of analyzing cyber-attack information received at various forms. World cyber-attack map and graph data visualization.

**Research evaluation of QUIC for optimal real-time video transmission (2016/3~2016/4)**

Evaluate Google’s QUIC protocol is a short span of 1 month to determine if it can be used for real-time video transmission protocol and transport feedback information that can be used to throttle video transmission to quick and rapid application response.

**Development of Cyber Physical Security (2015/10~2016/2)**

A web application for a task based work order entry system allowing/denying physical access into premises. System integrated controlled devices such as gate control (combination of card reader and face recognition) and a VMS intrusion detection system. The system developed in Java and policy generation and distribution is based on XACML standard.

**Development of Demo GUI for Resilient Platform (2015/8~2015/9)**

Developed a web application illustrating graphically data center computing resources.

Resource computation is done by a backend application that dynamically pushes new updates to client application. The client application developed using the Raphael JavaScript library.

**Image processing development for DMAT applications (2015/4~2015/7)**

Developed the receiver and player part of a remote image transmission system. The receiver part provided feedback bandwidth calculation results to sender that adaptively adjusts the encoding rate for optimum image transmission and quality.

**Cyber Physical Security (2015/7~2015/9)**

Worked on requirements specification and design concepts of a cyber-physical security system. The system integrates physical components like card reader and face recognition together with an integrated work order entry system.

**OpenDaylight controller development (2014/10~2015/3/30) nearly 6 months)** – Developed using Opendaylight’s model driven service abstraction layer a plugin for controlling IX-router openFlow capable switches.

**Network Function Virtualization Orchestration (2014/04~2014/09)** – Refined resource allocation algorithm applicable to NFV architecture. Also developed monitoring and control with such architecture.

**Distributed OpenFlow controller development (2013/10/09~2014/7/3)** – Participated in the ONOS research development project for six months.

**Network Function Virtualization development -** Developed a Proof-of-Concept lab design for various virtualization network functions comprising a vBRAS PPPoE subscriber system. This was a 3 months project with the main objective to provide all the groundwork to able to exercise and experiment with a VM resource allocation algorithm. In particular I implemented a messaging framework on the top of ZeroMQ that enabled applications to define any kind of messages using a schema expressed in JSON to exchange messages.

Taking careful consideration into the design of ZeroMQ proxy processes allows us to easily extend workers or clients if the workload required.

Further I developed mechanisms to support retrieval of virtual resources statistics essential to a resource allocation algorithm and other higher entities. Using the Chef Framework tool developed recipes to install service modules into virtual resources. Data persistence achieved using MySQL, and Redis cache is used to read saved data thus reducing the query rate significantly. To enable applications access to data in a same way that exchange messages using a schema I developed a data mapper process to be able to respond to all CRUD operations.

**OpenFlow development -** Designed and implemented the trema-edge open source (<https://github.com/trema/trema-edge>) project which is a trema controller framework supporting OpenFlow 1.3.x. The project spanned for 3 months and during that period I completed the development of the switch and ruby bindings. In this project I implemented switch’s protocol control module integrated together with switch’s datapath module. One of the challenging parts of this project was developing an event notification mechanism to be able to synchronize and transfer data between the two threads (protocol/datapath) without locking.

Apart from the added support of OpenFlow 1.3 trema-edge differs in many ways to trema. A big difference compared to trema is that trema-edge uses Ruby’s retrospection features to dynamically define all message objects. All constructed or received message objects can be inspected using Ruby’s inspect method which makes debugging and testing easier.

Full-time contributor to an open source framework for developing OpenFlow controllers using Ruby and C. (<https://github.com/trema/trema>). My initial role for this project was the development of the Ruby application interface layer (ruby bindings). My work also included research and development of lock-free algorithms to improve the execution speed of certain routines.

Documented and tested NEC’s quantum plugin API (<https://github.com/trema/apps/wiki/Sliceable-Network-Management-API>) that communicates with the Quantum controller in the OpenStack ochechestration platform.

In a project termed as Source Flow which embeds a list of actions into the user packet to avoid flow lookup on core nodes I developed the switch and controller modules required for this implementation. A relevant paper presented at the ACM SIGCOMM conference (<http://conferences.sigcomm.org/sigcomm/2010/papers/sigcomm/p465.pdf>).

Developed lab designs and test concept prototypes for a path manager that offered optimal path selection and balancing of flows across the network. In addition I developed a path resolver process that offered intelligent multi-path routing using OpenFlow 1.0.

In another project designed and implemented database synchronization code to ensure flow data is consistent across all nodes.

Debugged and identify fixes from OpenvSwitch's core dump files.

**Web application -** Designed and implement a web administration application used for the configuration of a VoIP system using Ruby on Rails (RoR) object relational model (ORM) with Representational State Transfer (REST) integration.

**Media control server -** Designed a media server resource management and distribution process to handle multimedia content for a VoIP enterprise application.

**Authorization, Authentication, Accounting (AAA) -** Led a team of 5 developers designed and completed the implementation of two Diameter protocols for Next Generation Networks (NGN) and IP Multimedia System (IMS) systems.

**Asset management system -** Designed and developed a web based wireless LAN asset management system using PHP and MySQL.

**Image processing** - Researched and developed an image processing system using stereo imaging to determine attributes of containers  such as number and height used for a cargo bay. Further developed a feature classifier based on generic algorithms for roman alphanumeric character recognition. Prototype developed using Java Advanced Imaging and ported to a Renesas imaging processing board.

NEC Corporation. Kamoi, Yokohama

Software engineer November 1998 - March 2002

**Low computational algorithms** - Developed and refined various channel coding and MPEG-4 transcoding algorithms for the Wide Band Code Division Multiple Access (W-CDMA) project. Results produced to meet the speed and code size of the constrained development environment. Project development carried out in C and DSP linear assembler and assembler.

NEC Australia Pty. Ltd. Mulgrave, Victoria

Software engineer April 1996 - October 1998

**Communication protocol** - Designed and developed three protocols (HDLC, LAPB/D, SNMP) for supervisory control and data acquisition used for a remote access digital subscriber microwave radio system. Project developed using C and C++.

A.E.S. Prodata Pty. Ltd. Prahran, Victoria

Embedded software engineer April 1995 - March 1996

**Communication protocol** - Designed and implemented a multilayer protocol for an automated fare collection system. Well defined interface specifications and my participation to a library group coordinating task activities among other teams resulted in the timely delivery of the project.

NEC Australia Pty. Ltd. Mulgrave, Victoria

Software design engineer March 1994 - March 1995

**Embedded software development** -  Developed embedded control software for a route diversity protection switching system. This project although small in size imposed strict real-time constraints in terms of execution speed, memory and program size with great emphasis on maintainability.

TabCorp Holdings Limited. Melbourne Victoria

Software engineer, Senior software engineer September 1990 - March 1994

**Communication protocol** - Designed and developed a fault tolerant TCP/UDP server capable of accepting and processing up a high volume of simultaneous client connections (maximum of 5000) for an electronic gaming system. The server was capable of switching processing load dynamically due to an observed stress load condition or to no response from the consumer process.

**Front-end reporting** - Designed and developed financial applications using Oracle and C/C++ for report reconciliation. Further integrate into the system statistical financial models to analyze takings.

**Project management** - At later stages of the project and after my promotion to a senior software engineer I involved in project effort estimation, project risk assessment and system tuning during testing.

**On-line support** - The project lead time from system concept to design, implementation and rollout was within a time span of just over two years. Once the system was released played a vital support role to ensure system remained operational 24 hours a day. Coordinating time to keep working on other duties as well as doing support proved to be a challenging task at that stage**.**

Logica Pty. Ltd. Box-Hill, Victoria

Programmer June 1989 - September 1990

**Billing system -** Developed alterations to reports used by the credit card tape generation and extraction charges modules. Incorporated error checking and audit trail error reporting for the end of day billing system. Work has been carried out using C and Advanced Database Systems Language.

Melbourne University Institute of Education. Melbourne, Victoria

Programmer November 1988 - June 1989

**Use-case analysis** - Involved in collecting and defining technical requirements for an equipment asset and terminal connections nodes database system. Requirements were documented using context diagrams that described the structure and behavior of the system domain. Potential use-cases were analyzed and verified against the system domain model.

**Education**

Monash University Melbourne Australia. Caulfield East, Victoria

Bachelor of Applied Science (Digital Technology) Feburary 1984 - November 1988

Short courses

* Texas Instruments DSP Programmers
* C++ Programming for C Programmers
* Hands on TCP/IP for C Programmers
* ORACLE 7 Database Administration
* ORACLE SQL, SQL\*Plus and PL/SQL
* VOS/System 88 Programmers
* VOS/System 88 Advanced Programmers
* VOS/System 88 System Architecture

**Foreign language skills**

Greek

Japanese Language Proficiency Test Level 3 (2002/02).

**Computer skills**

Languages: Ruby, Rake, RSpec, Cucumber, C, Java, C++, Visual C++, SQL, GNU Scientific Library, Texas Instruments linear assembler, assembler.

Web development: RubyOn Rails (RoR), PHP, HTML, DHTML, HAML, JSON, AJAX, Java Server Pages (JSP), CSS, Javascript, XML, XSLT, XPATH, XML-Schema, SMIL, DOM.

Revision control software: Git, SVN.

|  |  |
| --- | --- |
| **Skill** | **Amount of Experience** |
| Ruby | 7 years |
| Rake/RSpec/Cucumber | 5 years |
| Ruby on Rails(RoR) | 4 years |
| C | 20 years |
| Java | 12 years |
| SQL | 12 years |
| PHP | 5 years |

**Experience**

Experience with test-driven and agile development (cucumber, rspec, mocking, and fixtures).

Experience employing simple design and lightweight modeling techniques using TDD, BDD.

Experience managing off-shore development.

Experience with RDBMS (MySQL) and non-relational datastores (Memcache, MySQL(MyISAM)).

Experience with SOLID object-oriented design and programming.

Experience in designing and implementing network protocols.

Experience with OpenFlow based controllers and OpenFlow based switch development.

Experience with Layer 2 and Layer 3 switching.

Experience with event based socket IO paradigms.