Ontology-Driven Knowledge Based Autonomic Management for Telecommunication Networks: Theory, Implementation, and Applications

by

Qianru Zhou

Submitted for the degree of Doctor of Philosophy
at
Heriot-Watt University
School of Engineering and Physical Sciences

May 2018

Abstract

Current telecommunication networks are heterogeneous, with devices manufactured by different vendors, operating on different protocols, and recorded by databases with different schemas. This heterogeneity has resulted in current network managements system becoming enormously complicated and often relying on human intervention. Knowledge based network management, which relies on a universally accepted knowledge base of the network, has been discussed extensively as a promising solution for autonomic network management. To build an autonomic network management system, a universally-shared and machine interpretable knowledge base is required which describes the resources inside the telecommunication system. Semantic web technologies, especially ontologies, have been used for many years in building autonomic knowledge based systems in Artificial Intelligence. There is a pressing need for a standard ontology to enable technology agnostic, autonomic control in telecommunication networks. Network clients need to describe the resource they require, while resource providers need to describe the resource they can provide. With semantic technologies, the data inside complex hybrid networks can be treated as a distributed knowledge graph, where an SQL-like language – SPARQL is ready to search, locate, and configure a node or link of the network.

The goal of this thesis is two-fold. The first goal is to build a formal, machine interpretable information model for the current heterogeneous networks. Thus, we propose an ontology, describing resources inside the hybrid telecommunication networks with different technology domains. This ontology follows the `Device-Interface-Link` pattern, which we identified during the modelling process for networks within different technology domains. The second goal is to develop a system that can use this ontology to build a knowledge base automatically and enable autonomic reasoning over it. We develop a **S**emantic **E**nabled **A**utonomic management system of software defined **NET**works (SEANET), a lightweight, plug-and-play, technology-independent solution for knowledge-based autonomic network management that uses the proposed ontology. SEANET abstracts details of network management into a formally defined knowledge graph augmented by inference rules. SEANET's architecture consists of three components: a knowledge base generator, a SPARQL engine, and an open API. With the open API developed, SEANET enables users without knowledge of Semantic Web or telecommunication networks to develop semantic-intelligent applications on their production networks. Use cases of the proposed ontology and system are

demonstrated in the thesis, ranging from network management task and social applications.

To my parents: I will do whatever I can to make you happier every day.

Acknowledgments

Herein I acknowledge those people without whom I would never have finished this PhD course. I would like to thank my supervisor, Professor Stephen McLaughlin, not only for his distinct and accurate vision to always shade light on my path in the dark, but also for his kindness and grace which have saved me in more than one way. Special thanks go to Dr. Alasdair Gray. I am so very lucky to have his intelligent and patient guidance in the domain of semantic web and RDF stream processing. Without him I could never be able to complete my PhD work. I would also like to thank Professor Chengxiang Wang, without whom I would not have had the chance to study in such a beautiful city as Edinburgh.

During the three years working on the TOUCAN project, I have made many friends from whom I have received professional advice and support, especially during the ontology construction process. I would like to thank my colleagues in the University of Edinburgh, Lancaster University, and University of Bristol. Your expertise in networks of different technologies really helped me a lot.

I would not have been able to embark on an endeavour such as a PhD without the love and support of my family. My mother, Jinxia Wu, besides of her endless love and tolerance for me, she has never hesitated to provide support to me during my three years of living in the UK. I would also like to thank my boyfriend, Fuyong Yue, not only for taking care of me and providing suggestions as one who has been through it all, but also for being my source of joy during these years.

Qianru Zhou
Edinburgh, January 2018.

ACADEMIC REGISTRY
Research Thesis Submission

Name:	Qianru Zhou		
School:	School of Engineering & Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	

Submission

Submitted By *(name in capitals)*:	QIANRU ZHOU
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals)*:			
Method of Submission *(Handed in to SSC; posted through internal/external mail)*:			
E-thesis Submitted (***mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.
Academic Registry/Version (1) August 2016

Declaration of Authorship

I, Qianru Zhou, declare that this thesis titled, 'Ontology-Driven Knowledge Based Autonomic Management for Telecommunication Networks: Theory, Implementation, and Applications' and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at Heriot-Watt University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

Contents

List of Figures

List of Tables

List of Publications

Q. Zhou, S. McLaughlin, A. J. G. Gray, S. Wu, and C.X. Wang, "Lost Silence: An emergency response early detection service through continuous processing of telecommunication data streams," In *proc. ISWC Web Stream Processing workshop, Vienna, Austria, 21st-25th Oct. 2017.*

Q. Zhou, C.X. Wang, S. McLaughlin, and X. Zhou, "Network virtualization and resource description in software-defined wireless networks," *IEEE Commun. Mag., vol. 53, no. 11, pp. 110-117,* Nov. 2015.

C. Rotsos, A. Farshad, D. King, D. Hutchison, **Q. Zhou**, A. J. G. Gray, C.X. Wang, S. McLaughlin, "ReasoNet: Inferring Network Policies Using Ontologies," In *Proc. NetSoft 2018*, Accepted.

Q. Zhou, A. J. G. Gray, A. Farshad, C. Rotsos, D. Hutchison, C.X. Wang, S. McLaughlin, "SeaNet: Semantic Enabled Autonomic Management of Software Defined Networks," *J. Web Semantics 2018.* submitted.

Q. Zhou, A. J. G. Gray, S. McLaughlin, "ToCo: An ontology for hybrid telecommunication networks," J. Web Semantics, submitted.

Q. Zhou, A. J. G. Gray, S. McLaughlin, "SARA: Semantic access point resource allocation for hybrid wireless networks," *IEEE Comm. Lett.,* submitted.

Chapter 1

Introduction

1.1 Background

The explosion of telecommunication services has resulted in today's network infrastructure becoming increasingly dense and complex [4]. This increasing complexity has led to even more complex network management. Current network management functions and infrastructure are overly complex. They are defined by a plethora of acronyms, from protocols (e.g., OSPF, BGP, MPLS, LDP, RIP, PIM, and IS-IS, etc.) to applications (e.g., HTTP, HTTPS, SNMP, SMTP, MIB, FTP, POP, RTP, and RTCP, etc.), from enterprise networks (e.g., VLAN, ARP, VTP, NAT, MAC, and DHCP, etc.) to wireless networks (e.g., LTE, eNodeB, 3GPP, UMTS, RRC, HLR, VLR, GRPS, UTRAM, CDMA, and CD/CAICH, etc.), and this is not a complete list [5]. State of art network infrastructure is also enormously complex, there are routers, gateways, bridges, router servers, switches, firewalls, and NATs, etc. For traffic control, there are packet shapers, packet sniffers, scrubbers, load balancers, etc. Most of these devices differ from each other in subtle ways. To compound motives, there are various operators and equipment vendors, e.g., Huawei, Samsung, Three, O2, Cisco, Erisson, etc., who construct their own mechanisms and have their own versions of configuration, description documents, technical specification, and software systems for devices with a similar function. The current technical specifications and standardisation documents of networks are also problematic [5], [6]. Multiple solutions and standards exist with different approaches to the same problem. For example, there are a large number of competing The Internet Engineering Task Force (IETF) Requests For Commentss (RFCs) providing solutions to similar questions.

This situation cannot continue. Uptime, reliability, and predictable latency requirements drive the need for automation. As noted by Shenker, the ultimate goal of the Software Defined Network (SDN) is to find appropriate abstractions, and break the complex network management tasks into tractable elements [4]. The key to network abstraction is the development of a highly abstracted information model which is universally accepted and machine interpretable [7], [8]. Thus, the policies and decisions made at a higher level can be translated into appropriate lower level operations automatically and hide the technical details from the network management system. This will spare humans from the complicated, low-level jobs such as device configuration [9]–[15]. Hence, the first question that this thesis tries to answer is: *"is it possible to build an information model describing all of the information of a heterogeneous network, which is universally accepted by all the technology domains and is machine interpretable?"*

Current information models for telecommunication networks management are problematic. Most of them are defined for a specific protocol, focusing on a single network layer. As can be imagined, out of the scope of the protocol, or when the protocol is replaced by a new one, these information models are not very practical [16]. Take YANG, a data modelling language proposed by network configuration protocol (NETCONF) for example, there are YANG modules proposed for WiFi[1], IP management, hardware management, etc, separately [17]–[19]. These YANG modules will provide a view of the resources in the device from different aspect. However, they could not provide an universal view of the resources technology independently.

Rooted in artificial intelligence, knowledge-based system have been extensively investigated in developing autonomic systems [20]–[35]. Generally speaking, it is a computer program that can represent and infer knowledge by means of ontologies and

[1] https://github.com/openconfig/public/tree/master/release/models/wifi

rules. The operation of these systems relies on three components: the knowledge base, the inference engine, and the interfaces. A knowledge-based system requires formally structured data for its knowledge base, not only databases with numeric and literal records, but also pointers direct to other objects which in turn have more pointers. In other word, these data are linked [36]. The ideal representation for linked data is an ontology. Semantic web technologies [37] can be used to present computer-understandable semantics from unstructured datasets through formal and consensual terminologies. The ontology driven knowledge base is the answer to the first research question. Thus, the ultimate task of network abstraction turns into trying to solve the problem, *can we build a knowledge-based network management system* and *what applications can be developed based on it*, which is the second and third questions that the thesis tries to answer.

1.1.1 Scope of Thesis

The aim of this thesis is to *develop mechanisms to enable technology-independent, autonomous management of telecommunication networks with hybrid technology domains, based on a universally accepted and machine interpretable knowledge base.* To achieve this goal, the following objectives are established.

- **Investigate current technologies:** The current problems in telecommunication network management are analysed along with the solutions proposed to date. Technologies that have been adopted to solve the problems are also analysed in detail. The results of this investigation will be used in our system design and implementation.

- **Develop a knowledge base:** Summarizing the drawbacks and advantages of current knowledge bases, propose our knowledge base for hybrid telecommunication networks. Knowledge from domain experts of each network technology domain are translated into a machine interpretable knowledge base.

- **Design a system to implement the knowledge base:** A system acting as the wrapper for the telecommunication networks – knowledge base will be designed. It should be able to:

  1) generate the knowledge base automatically for networks adopting any kind of technology or with any topology;
  2) execute operations on the knowledge base, e.g., query, reason, add, delete;
  3) perform the basic network management tasks;
  4) provide an open technology-independent interface, exposing the network management operations to users without knowledge of Semantic Web technologies or telecommunication networks.

- **Implement the system in emulated networks:** On telecommunication networks emulated based on real networks, a prototype of the designed system will be implemented on selected platforms. The extension on different platforms will also be considered and demonstrated.

- **Evaluate the implementation with use cases:** Serval use cases with different focuses and tasks will be carried out to evaluate the system proposed. The evaluation experiments will be designed and carried out based on real life scenarios. The performance of the system as well as the overheads will be analyzed and compared with current network management systems.

1.2 Contributions

This thesis makes the following key contributions towards an ontology driven knowledge based autonomic network management system.

Literature Review

- Reviews recent advances on knowledge based autonomic network management, and classify recent advances on existing ontological models for telecommunication networks with hybrid technologies.

Research on ontology engineering for telecommunication network with hybrid technologies

- Proposed a novel ontology for telecommunication networks with hybrid technologies – the TOUCAN Ontology (ToCo) [38]. The requirements specified for the ontology of autonomic network management are studied in the form of competency questions. The reason for the research gap of current ontologies are discussed. The key modules of ToCo are described from five perspectives in detail.

- For the first time to our knowledge, an ontology design pattern for networks is proposed – the `Device-Interface-Link (DIL)` pattern. It is observed and summarised during the constructing process of ToCo. The `DIL` pattern identifies and provides an important insight into the abstract and recurring knowledge structure in the domain of networks, or more generally, any kinds of linked things. With the `DIL` pattern, the ontology modelling processes for networks are made clearer and efficient.

- Perform ontology evaluations for ToCo, by developing several concrete examples and use cases. It has been demonstrated that ToCo is able to model various kinds of telecommunication networks, from small-scale telecommunication networks such as vehicle-to-vehicle networks and smart home devices, to large-scale networks such as satellite networks.

Research on knowledge base autonomic network management

- Design and develope a knowledge-based autonomic network management system. The architecture collects network information automatically into a knowledge base with the assistance of the ToCo ontology, and provides rule-based inferencing. To the best of our knowledge, it is the first open source knowledge based autonomic network management system driven by an ontology. We take a modest step towards the realisation of semantic SDN autonomic management in practice.

- A mechanism to accomplish a knowledge-based autonomic network management system for SDN. This mechanism consists of three components, a knowledge base generator, a rule-based inference engine, and a Linked Data Application Programming Interface (API). The knowledge base generator is designed to collect unstructured information from the network elements, e.g., the switches, hosts, ports, and flows in each switch, then to map them into a structured, ontology-assisted knowledge base, which can be processed directly by computers. The knowledge base generated is ready for semantic queries, which are carried out by the inference engine. The Linked Data API is provided as the interface between underlying technology-specific operations and the high-level technology-independent management. A handful of network management rules are provided as functions.

- Several real-life use cases implementing the knowledge based autonomic network management mechanism, namely, learning switch, firewall, resource allocation. The approach has also been shown to be beneficial for a disaster response scenario [39].

1.3 Thesis Organisation

This thesis is organised into 7 chapters as follows.

Chapter 2 discusses research on the key technologies adopted in this thesis. It begins with a literature review of the current research on knowledge representation for networks and their limitations. Then the chapter goes on to discuss Artificial Intelligent technology – knowledge engineering, which is believed to be a promising solution to overcome these limitations. The key concepts and the methodology to develop a knowledge base are discussed in detail. The semantic technology, especially ontology, are highlighted. The rest of Chapter 2 discusses the SDN, which is the future of telecommunication networks. The architecture and current control methodologies are discussed in detail. Its advantages comparing with the traditional networks are highlighted. The software tools adopted in this thesis for both knowledge engineering and SDN are also introduced at the end of this chapter.

The next 4 chapters present my solutions towards the goals of the thesis. Chapter 3 presents the detail of the development process for our proposed ontology, ToCo. The physical infrastructure of the hybrid telecommunication networks, including devices, interfaces, and links in both wired and wireless networks are described in ToCo, as well as the quality of communication channel, such as bandwidth, data rate, package loss, delay, etc. Finally, concepts of services provided by the telecommunication networks, as well as customers, are also included, as they are part of the telecommunication system. The chapter begins with the competency questions ToCo must satisfy, the modules in ToCo, as well as the DIL pattern on which ToCo is constructed. The chapter then goes on to evaluate the ontology by examples and use cases. Three network resource description examples for Wireless Fidelity (WiFi), Light Fidelity (LiFi), computer network, respectively and a SDN flow description example are presented. Three use cases of ToCo are introduced. The related ontologies literature review is presented at the end of the chapter.

Chapter 4 presents our proposed knowledge base autonomic management system for telecommunication networks – Semantic Enabled Autonomic Management of Software Defined Networks (SEANET). The chapter begins with the development environment, both the software and the hardware, and the architecture. There are three key components in the architecture of SEANET: a knowledge base generator, SPARQL engine, and a Linked Data API. The knowledge base generator can collect resource descriptions from the network, and by mapping them to the ToCo ontology, a knowledge base is generated for the network. The SPARQL engine can run queries on the generated knowledge base, and any inference on it. A technology independent API is provided to make the functions of SEANET accessable for users without knowledge of either telecommunication or knowledge engineering. These functions range from knowledge base generating, query executing, to network management task executing. The given evaluations allow us to prove the practicability of SEANET as an autonomic network management system.

The following two chapters, Chapter 5 and Chapter 6, give details of two projects as the implementations of the proposed SEANET system and ToCo ontology. Chapter 5 investigate the strategy of access point selection in a hybrid network. In this network, there are multiple access points with different technologies available for each user. The strategy will be able to select the best access point that could provide the optimised communication service to the user. Chapter 5 begins by introducing the current assess point selection strategies and their limitations. Then our proposed autonomic, semantic-intelligent solution implementing SEANET is presented in detail, including the steps of our proposed strategy, the algorithms, SPARQL query, and the rules adopted. Finally, we evaluate it in hybrid networks with different scales, ranging from 10 users to 100 users. The evaluation results are collected and compared with current strategies, with the metric of the bandwidth. The overhead in terms of execution time is also collected and investigated.

In Chapter 6, we demonstrate an implementation of SEANET in the domain of disaster response. With the semantic knowledge base provided by SEANET, the data inside the telecommunication networks can be conveyed and analysed automatically. The use case – lost silence presented in Chapter 6 demonstrates a methodology to detect shipwreck accidents immediately (with the delay in the order of milliseconds), by processing semantically annotated streams of data in cellular telecommunication system. With SEANET, real time data of phones, such as position and status, are encoded as RDF data streams. In Chapter 6, we first present the motivating scenario and the background technologies, including RDF stream processing and the location registration mechinism in cellular network. Then the algorithm that by processing streams of RDF annotated telecommunication data to detect abnormality. Finally, we exemplify our approach with a real shipwreck accident – “Eastern Star” in Hubei China. Our evaluation results show that with a properly chosen window size, ship capsizing accidents can be detected effectively.

Finally, Chapter 7 presents the conclusion of thesis, and suggests possible future research directions as extensions of this thesis. It is suggested that, with the structured knowledge base generated in the thesis, we could use machine learning and stream process to further implement the data inside the telecommunication system.

Chapter 2

Knowledge Representation for Software Defined Networks: A Literature Review

2.1 Introduction

In this chapter, research topics relevant to this thesis will be introduced with current research result in the literature. This thesis touches on the areas of knowledge engineering, semantic web, software defined networks and policy-based network management.

The rest of this chapter is organised as follows. In Section 2.2 we first look at the knowledge representation for networks, the current methodologies for representing knowledge are introduced and their limitations towards an autonomic network management system are discussed. In Section 2.3 the background of knowledge engineering, e.g., basic concepts, technology details, and the key technology adopted in this thesis – Semantic Web technologies are discussed. The reason for adopting ontologies to represent network resources, the state of art of current ontologies proposed for telecommunication networks, and the approaches to implement ontologies (through ontological applications) are discussed in detail. Section 2.4 brings out the future of networks – software defined networks (SDNs), the control strategy and the current autonomic management approaches. Adopting the knowledge engineering technologies to represent the SDNs, Section 2.5 proposed the concept of Semantic Knowledge Based SDN. First, the concept is illustrated in a meta-modelling perspective, then a transition from syntax to semantic knowledge base building is discussed. The survey of software tools available for knowledge engineering and SDN developing is illustrated in Section 2.6. The current research gap is summarised in Section 2.7, followed by a conclusion Section 2.8.

2.2 Knowledge Representation for Autonomic Network Management

2.2.1 Autonomic Network Management

Network management is the process of maintaining, configuring, and monitoring the telecommunication networks. The software systems that implement network management are called network management systems. An autonomic network management system is an autonomic system which can perform the network management control tasks without intervention from external software or a human. It should also be able to close the control loop and adapt to the changing environment by itself [40]. An autonomic managed network should possess the following self-properties [40]:

- ***Self-configuration:*** Based on the self-knowledge retrieved by the network on its own, and high level policies defined by admin, the network should be able to configure itself, without the intervention from humans or external softwares.

- ***Self-healing:*** Networks adapt themselves to the changing environment, recover themselves from function failures.

- ***Self-optimizing:*** Networks automatically make decisions to optimise their performance towards a set of well-defined goals.

- ***Self-protection:*** Networks automatically secure themselves against internal and external attacks.

A knowledge based system, rooted in Artificial Intelligence, has long been used to accomplish an autonomic system. The key to a knowledge based system is a semantic intelligent, cryptographically verifiable knowledge base of the network and is the cornerstone for the autonomic network management. SDN is proposed to accomplish an autonomic managed network, by providing an abstract control plane.

2.2.2 Current knowledge bases for Network Management

This section will shine light on the current knowledge bases adopted by the telecommunication networks to describe network resources, mainly the topology information.

2.2.2.1 Simple Network Management Protocol

The simple network management protocol (SNMP) is part of the Internet protocols defined by Internet Engineering Task Force (IETF) [41]–[48]. It supports the network management system by providing watching and monitoring network devices in case of any incidents [1]. SNMP consists of a set of network management standards, including an application layer protocol, a database schema, and the data objects. SNMP is implemented by a computer program running in the systems of the devices managed, namely, an SNMP agent, which can expose the information of the device as variables. These variables defined by SNMP are organised in tree-based hierarchies (an example of the tree-based variables hierarchy of MIB is given in Fig. 2.1). Each variable is assigned with an identification – object id (OID). Together with the other metadata, these hierarchies are known as the management information base (MIB) [1], [41]–[48].

Although it is proposed for network configuration, SNMP is mainly adopted for fault and performance monitoring rather than device configuring. This is mainly due to its complex data structure and the lack of a universally accepted standard for a

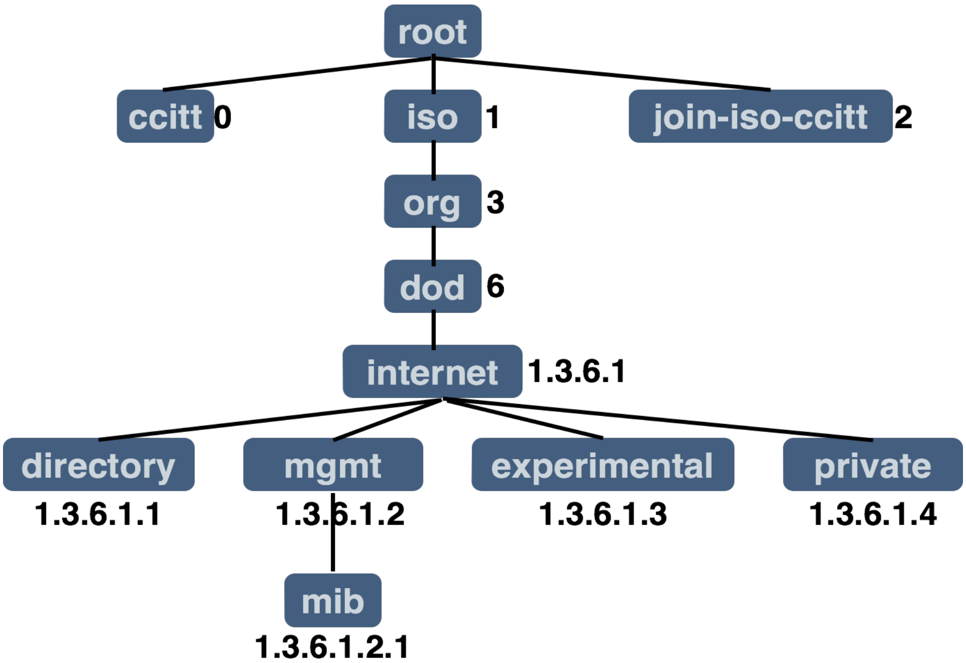


FIGURE 2.1: Tree Based Hierarchy in MIB [1].

certain technology. For example, different equipment vendors (Cisco, Huawei, Juniper Networks Inc., Ericsson-LG, Samsung, etc.) tend to define their own MIBs for devices accomplishing the same features. These MIBs are neither open nor shared. Thus, it is very time and labour consuming to develop an autonomic configuration process choosing the correct MIBs for all the devices.

2.2.2.2 Command Line Interface

Although SNMP has proven to be a very popular network management protocol, not every network equipment vendor supports it. Thus, in 2002, the same organisation that developed SNMP, IETF, drafted a proposal for Command Line Interface (CLI) [49]–[52]. Unlike SNMP, CLI is a text-based approach to autonomic configuration to devices in a network. With all its features unfortunately, CLI has various limitations and shortcomings. One of the most significant disadvantages is that it does not support transaction management [43], [50]. The configuration of a device usually requires multiple steps, and sequential cooperation from different devices. Thus, once one step fails, we have to rollback, i.e., undo all the previous operations. This will require huge amount of complexity in programming with a network configuration protocol does not support transaction management, such as CLI [43]. CLI cannot provide standard format for output or error information, which is another important issue that prevent autonomic network management.

2.2.2.3 Network Configuration Protocol

It has been estimated that up to 60% of configuration errors are due to humans typos or translating errors from print to their console [43]. To save the humans from the tedious configuration loop, network configuration protocol (NETCONF) is proposed to address the shortcomings of all the current configuration and management protocols [43]. It defines operations to install, manipulate, and delete the configurations for devices in the network. NETCONF also defined an XML based data modeling language – YANG [53]–[55]. Associating each term with a unique URI, YANG describes the data of the network in a structured way. It is hoped that, with the emergence of SDN, NETCONF can replace SNMP and CLI as the configuration management for programmable networks [43].

2.2.3 Limitations of the Current Network Information Models

Although various protocols and technologies have been developed to model for network management, they failed, in one way or another, to be an efficient technology to satisfy the general needs of autonomic network management [9], [12], [14]–[16], [20], [21], [41], [52], [55]–[68]. The reasons are summarised as follows.

- **Limited to Scalar Variables:** current information modelling capabilities are limited to scalar variables [15]. The main constraints of these information models are that they do not have the relations, constraints, semantics, and axioms on their data, which could enable autonomic reasoning and inferencing [62].
- **Lack of Transactions:** Current wireless communication networks are heterogeneous with different access technologies, e.g., wireless fidelity (WiFi), light

fidelity (LiFi, a wireless communication uses light to deliver signals), radio frequency (RF), etc., working together to provide better capacity and enhance energy efficiency of the system. However, the existing information models have only partial models of the network, and are constrained to each specific technology domain. The transaction capabilities are quite lacking, let alone to be used for the automation tasks in order to meet the requirements of the ever growing network [15].

- **Limited Function:** almost all the current management protocols are limited to the network configuration only.

- **Lack of Semantics:** the current network models are semantically empty. Their ability for expressing the information model are not strong enough to provide sufficient semantics for automatic network management.

- **Lack of Global View:** almost all the current approaches offer a limited description of neighbour nodes and cannot provide real-time updated information model for networks.

In summary, the network management system has became the bottleneck of any further advancement towards the intelligent networks. To achieve the semantic intelligence, a technology in the domain of artificial intelligence, knowledge engineering (or its core technology, ontologies), has been increasingly used in network management, which could allow machine interpretability and management. The following section will give a detailed introduction to this technology and the mechanism of how to build a knowledge base with ontology – the formal structured, machine interpretable model for telecommunication networks.

2.3 Knowledge Engineering

2.3.1 Basic Concepts

Knowledge and reasoning play a crucial role in the field of Artificial Intelligence. A knowledge base is the knowledge of a specific domain represented by the concepts and relations formally formatted by a knowledge description language [69], [70]. The process of constructing a knowledge base is called knowledge engineering [6], [22]–[26], [28], [29], [35], [56], [69], [71]–[79]. A knowledge-based system is a software solution that adopts and inferences on a knowledge base to solve problems in a domain [6], [70], [72]. There are three components in a knowledge-based system: a knowledge base, an inference engine, and a user interface. Central to any knowledge-based system is the knowledge base, which contains all the domain's knowledge as a collection of facts, or in other words, all that is known in a domain. The approach to store knowledge into a knowledge base is called knowledge representation [24], [25], [27], [70]. The ideal representation for a knowledge base is the ontology [27], [70]. An ontology together with the instances it generated form the knowledge base. With the assistance of the ontology, the system can achieve competency swiftly by being told or learning new knowledge of the certain domain. For example, in the network management domain, the knowledge-based management system can adapt to the changing environment of the communication system by updating the knowledge base [20]–[35]. An inference engine is able to adopt the existing knowledge base to answer customised questions and come to the conclusions.

Artificial Intelligent research mainly focuses on how to develop an operational knowledge-based system. In the early 1980s, researchers of Artificial Intelligence think knowledge-based system is realised by transmitting human knowledge from the brain of human experts into a computer processable knowledge base [25]. However, this paradigm has been proven to be problematic [6], [21]–[26], [28]–[35], [69], [71], [72], [75], [79]–[83]. Due to the fact that it is difficult to separate the problem to be solved from the causes of the problem in the tacit knowledge or experience from human experts. Nowadays, knowledge based system building process has developed into a modelling process. Building a knowledge based system is equivalent to building a computer processable model with the capability to solve problems parallel with the human experts [6], [21]–[26], [28]–[35], [69], [71], [72], [75], [79]–[83].

2.3.2 Steps of Knowledge Engineering

The procedure of knowledge engineering varies greatly with domains. However, the basic steps are as the following [6], [22]–[25], [27]–[35], [69], [71], [72], [75], [80], [81], [83], [84].

- ***Define the domain:***
  It is an impossible mission to represent everything in the world, although it is the ultimate goal of knowledge engineering. Usually knowledge engineering investigates a specific domain whose circumstance is delicately defined and the range is known in advance [30]. Even for the same object, different perspectives can lead to vast differences in the developed knowledge base. Take a simple wired computer network for example, if we are building a knowledge base for network engineers, the facts we are interested are probably the IP address, bandwidth, data rate, packet loss, etc. However, if we are building it for the maintenance staff, the facts of the network that interest us are probably, the model type of the server, the service life of the cable, the temperature of the datacenter, and the air humidity, etc.

- ***Acquisition knowledge:***
  The knowledge engineer should be working with the domain expert, or, quite conveniently, be a domain expert himself/herself. In this stage, knowledge needs to be extracted from the head of the experts in the domain, which could be anything they know, e.g., industry knowledge, professional skill, or personal experience. During our knowledge engineering process for this thesis, we are working for the project TOUCAN[1], which has four Universities[2] with their own

[1] EPSRC TOUCAN project, No. EP/L020009/1. Website: http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/L020009/1

[2] University of Bristol, University of Edinburgh, Heriot-Watt University, and Lancaster University

expertise in different technology domains of telecommunication networks. Thus, we have experts from several network technology domains to work side-by-side together on the knowledge base.

- ***Enumerate key terms and rules in the domain:***
  This stage is also known as developing the vocabulary for the domain, which is translating technology-specific concepts into logic-level, formally defined names. It involves many detailed questions, e.g., should the links between hosts be represented as an object or by a unary predicate, should the status of a phone be a predicate or a function, etc. The finally decided vocabulary is the **ontology** of this domain.

- ***Define a description of facts:***
  This step pins down the axioms of all the concepts in the ontology and allow the experts to check the content. The previous step and this one form an iterative process, which are executed until the ontology is perfected. However, it is worth mentioning that there is no such thing as "a perfect ontology", because there is always room to improve the ontology, and with the development of the technology, the ontology should be updated and/or expanded as well.

- ***Execute the inference procedure:***
  Once the ontology is properly defined, we can feed the axioms and facts about the problems into the inference engine, and get the results.

- ***Debug the knowledge base:***
  Just like developing the ontology, building a knowledge base is an iterative process. There is no "perfect" knowledge base in the world. Normally, the knowledge base is built to fulfil certain use cases in the domain. Thus, the best way to evaluate it is during the implementation process in the use cases. However, it should also be able to adapt to new technologies and new use cases that pop up in the future. Thus, the knowledge engineer needs to make the knowledge base extendable for future uses when developing it [77], [85]–[87].

2.3.3 Semantic Web

Berners-Lee proposed the Semantic Web with the aim of creating a universal medium so that data can be shared and processed by machines and people [88]–[95]. The goal of the Semantic Web is to define a universally recognised model, which can be directly adopted by machines to process and correlate the information. An ontology is such a tool, defining the data for technology-independent applications and software agents to use. Semantic Web technologies investigate how to represent knowledge. It aims to how to enable computers to know things and do reasoning like humans do, which is the core target of the entire field of Artificial Intelligence. Knowledge representation studies how to use first-order logic to represent the facts about the real world to computers, such as time, location, actions, etc. [30]. For example, in the domain of network management, ontologies, are adopted to represent knowledge about network topologies, physical infrastructures, port status, traffic, channel capacities, etc. Semantic web technologies consist of resource description framework (RDF), and ontology.

2.3.3.1 Ontology

In the field of artificial intelligence, the ontology is known as the vocabulary, defining a set of classes and the relations between these classes in a domain, in a formally defined and universally agreed structure, that can be communicated between people and machines [12], [26], [29], [34], [69], [77], [79]–[81], [85], [88]–[102]. The process of representing knowledge with abstract concepts is called ontology engineering, which is a more general term of knowledge engineering. The ontology is developed for knowledge sharing and reusing. An ontology together with its instances is the knowledge base of the domain. Take the network management domain for example, the ontology describes a set of nodes and the links between them. It is a consensual, shared, formal description of the knowledge in network management domain. The resource domain here is defined by physical infrastructure and logical entities in networks, such as switches, routers, devices for computation and storage, and the links between them, etc.

Based on their content and target domain, the ontologies can be divided into three levels, the upper level ontology, the lower level ontology, and the application ontology [10], [27], [29], [35], [80], [81], [85], [103], [104]. Strictly speaking, the application ontology is not really an ontology, because it is not shared and simply describes the resources for a specific application [88], [89], [92], [99], [103]. The upper level ontology, or core ontology, should not be too large, because its associated sub-ontologies will be based on it and will inherit its classes and properties [104], [105]. It describes the basic relations between the entities in the resource domain. When defining the classes of a core ontology, we need to find the most general class of this particular domain. The upper level ontology bridges the gap between different domains. The lower level ontology, also known as the fundamental ontology, contains everything else. Normally, the lower level ontology is not exposed, because they contain detailed information about the users and devices. For example, in Table 2.1, the "OpenMobileNetwork (OMN)" ontology described the resources in the WiFi domain with 32 classes and 51 properties [106].

2.3.3.2 Scope of Ontology for Autonomic Network Management

The scope specified for the ontology of autonomic network management are listed below in the form of Competency Questions (CQ). CQ are user-oriented interrogatives to define the scope of the ontology. In general, they are questions that the ontology and its associated knowledge base are designed to answer [80], [107], [108].

The principal competency question for the autonomic network management ontology is:

CQ0 Given the requirement for a particular type of service, what data can the telecommunication systems provide, where is the data they stored, and how often is it updated?

This principal competency question can be refined into more detailed questions. Some examples are shown in the Appendix A. In order to derive these questions, we consulted experts from various technology domains, taking the scope of the autonomic network management into consideration.

2.3.3.3 Existing Ontologies for Autonomic Network Management

Ontologies that have been proposed for network information modelling are numerous in the literature [9]–[15], [105], [106], [109], [110]. The most popular ones are summarised below.

Network Description Language [16]: Network Description Language (NDL) is the first description language to describe computer networks. It provides several sub-ontologies that can be used for that purpose: a *topology sub-ontology* that describes the basic interconnections between devices, a *layer sub-ontology* to describe technologies, and a *capability sub-ontology* to describe network capabilities, a *domain sub-ontology* for creating abstracted views of networks, and a *physical sub-ontology* that describes the physical aspects of network elements, like the blades in a device [16].

Ontology for 3G Wireless Network [9]: This ontology is proposed for wireless network transport configuration. It consists of two sub-ontologies, domain ontology and task ontology.

Mobile ontology [10]: Proposed for the SPICE Project[3], the mobile Ontology makes a great effort for ontology standardisation [10]. It is proposed as a scalable solution with several pluggable sub-ontologies, services, profile, content, presence, context, communication resources sub-ontology.

Ontology for Optical Transport Networks (OOTN) [105]: OOTN is a lower-level ontology for optical transport networks based on ITU-T G.805 and G.872 recommendations. It is a computational optical ontology [105].

[3] https://cordis.europa.eu/project/rcn/80707_en.html

Ontology adopted in "OpenMobileNetwork" [106]: "OpenMobileNetwork" is a linked Open Dataset for Mobile Networks and Devices. It also developed an open source platform that provides semantically enriched mobile network and WiFi topology resource in Resource Description Framework (RDF) [106]. The ontology it adopts is published online[4] and is efficient and mature for the description of mobile network topologies. However, that also limits the ontology to the specific scenario. For example, it cannot describe optical backbone networks or LiFi.

Existing ontologies proposed for network resource with various structures are summarised in Table 2.1. Table 2.1 lists the ontologies with and without layered structures. The ontologies with layers, listed in the upper part of Table 2.1, aim to describe overall network resources, e.g., mobile ontology [10], SOUPA ontology [111] for mobile networks, and network description language (NDL) [16] for optical networks. The lower-level ontologies are proposed for some specific scenarios, rather than the whole network. For example, in Table 2.1, OMN is a linked open dataset for mobile networks and devices [106]. This open source platform provides a semantically enriched mobile network and WiFi topology resource. Because of this, the OMN ontology is limited to that specific WiFi domain, e.g., it has not taken into consideration a phone's status.

Adopting ontologies and rules to represent knowledge has several advantages, comparing to conventional databases, as shown below.

- *Machine Programmability:* The ontologies and rules can provide a formal and well-structured representation for resources and logic, which could describe domain resources, client requirements, and tasks in a machine programmable way.
- *Autonomic:* Once the ontology developed for telecommunication networks, rules and policies can be directly instantiated and verified by devices in the form of software.

[4] http://www.openmobilenetwork.org/ontology/

TABLE 2.1: Existing Ontologies With Layered Structures Proposed for Network Resource Description.

Name	**Description**	**Layered Structure**
Network Description Language (NDL) [63]	The first description language to describe optical networks	**Topology schema:** describes the basic interconnections between devices **Layer schema:** describes technologies **Capability schema:** describes network capabilities **Domain schema:** creates abstracted views of networks **Physical schema:** describes the physical aspects of network elements, like the blades in a device
Wireless Ontology for 3G communication network [9]	It was proposed to solve a complex real-world network configuration problems.	**Domain** ontology **Task** ontology
Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) [111]	SOUPA is designed to model and support pervasive computing applications.	It contains two parts of ontologies: **SOUPA Core Ontology** and **SOUPA Extension Ontologies**. SOUPA Core Ontology: **Person, Policy, Agent, Time, Space, Event** SOUPA Extensions Ontology: **Meeting & Schedule, Document, Image Capture, Region Connection Calculus, Location**
Mobile Ontology proposed for SPICE project [10]	A standardized ontology that describes semantic models for scalable NGN service delivery platforms	**Core Ontology** **Service** sub-ontology **Profile** sub-ontology **Content** sub-ontology **Presence** sub-ontology **Context** sub-ontology **Communication Resources** sub-ontology
Ontology for Optical Transport Networks (OOTN) [105]	A computational optical ontology for optical transport networks based on ITU-T G.805 and G.872 recommendations.	
“OpenMobileNetwor ontology (OMN ontology) [106]	It provides approximated and semantically enriched mobile network and WiFi access point topology data.	No layered structure.
Energy Description Language (EDL) [112]	Describes energy resource for base stations in wireless networks.	

- *Linked Data:* Adopting ontology to represent knowledge allows not only the concepts, but also relationships between concepts to be mapped, . Thus, first-order logic can be adopted for rules and policies constructing and reasoning.

- *Technology Independent:* Adopting rules means domain experts can edit and revise the rules themselves rather than via computer programmer [36].

- *Self-Explanation:* With the knowledge represented explicitly with rules in the knowledge base and the inference engine, the system can reason about the domain knowledge and develop conclusions in a self-explainary way.

- *Universally accepted, formally defined, machine interpretable format.*

2.3.3.4 Ontology Evaluation

The evaluation of an ontology is problematic. Although many evaluation theories have been proposed [85], [86], there are few reports that describe the evaluation in detail in a step-by-step basis. Only very few ontology-based network applications have been developed and published. Thus, it is difficult to determine if one particular ontology is superior over any other. The mobile ontology proposed for the SPICE Project [10] attempted ontology standardisation. However, there is no reported evaluation of it.

The criteria for performance evaluation are consistency, completeness, conciseness, expandability, and sensitiveness [85], [86].

Consistency means every definition in the ontology is consistent and no contradictory conclusion can be deduced from other definitions and axioms. Ontology is consistent if and only if every one of its definitions is consistent.

Completeness is the basic requirement of ontologies, thus, incompleteness is a fundamental problem in ontologies [85], [86]. Completeness of an ontology is difficult to prove, however, we can prove the incompleteness of the ontology by proving the incompleteness of an individual definition.

Conciseness can be obtained for an ontology if and only if the following requirements are met:

- The ontology does not contain any unnecessary definitions or axioms;
- There are no redundancies between definitions;
- No redundancies can be inferred from other definitions and axioms.

Expandability means that new definitions can be added to ontology in the future without changing the already defined properties.

Sensitiveness is defined by the smallest change in a definition that can affect the already defined properties of the ontology [85], [86]. Broadly speaking, evaluation can be divided into two parts: technical evaluation, which is carried out by developers, and users evaluation. From another perspective, the ontology evaluation approaches contain two aspects: ontology verification and ontology validation [85], [86]. Ontology verification means that the ontology should be built correctly, which means its definition, in the natural language from the real world, matches the ontology requirements and competency questions of the target resource domain precisely. Ontology validation means that the ontology model matches the resource source in the real world correctly. Besides, ontology assessment is to judge the understanding, usability, usefulness, quality and portability of the definitions, taking the stand as the users. Various users and applications need various approaches to assess ontology [85], [86]. The target context for ontology evaluation comprises the following aspects [85], [86]:

- Each individual definition and axiom;
- All the groups of definitions and axioms that are stated explicitly in the ontology;
- Definitions imported from other ontologies;
- Definitions and axioms that can be inferred from other definitions.

In a nutshell, many reports have been published introducing ontology evaluation. However, very few offer details about exactly how the ontologies are evaluated or how the evaluation tools are built for different ontologies. It is worth mentioning that, just like software testing, ontology evaluation should be performed as early as possible during the developing process of the ontology and should be carried out throughout the entire life circle of this ontology [85], [86].

2.3.3.5 Resource Description Framework

The RDF is the format standard for ontology files [29], [92], [113]–[116]. Each resource (a concept or relation) is defined with a unique id, in the form of Uniform Resource Locator (URL) or International Resource Identifier (IRI). To simplify the URL/IRI, RDF adopts prefixes as shortcuts to represent the URL/IRI namespaces for the resources. For example, a resource represented with the URL/IRI as

http://purl.org/toco/switch1

can be simplified into "`net:switch1`" by using the prefix "net" to replace the namespace:

net: "http://purl.org/toco/"

RDF describes resources in the form of triples. A triple is an expression in the form of "subject-predicate-object". A set of triples constructs a graph [37], [75], [89], [92], [94], [95], [99]. Examples of RDF triples are shown below, adopting the prefixes given above.

```
net:switch1 net:hasPort net:port1.
net:link1 net:connectTo net:port1.
net:link1 net:connectTo net:port2.
net:host1 net:hasPort net:port2.
net:switch1 pos:locate pos:Point_sw1.
pos:Point_sw1 pos:longitude "123.001".
pos:Point_sw1 pos:latitude "321.001".
```

The `net:switch1`, `net:port1`, `net:link1`, and `pos:Point_sw1` are concepts, and the `net:hasPort`, `net:connectTo`, `pos:longitude`, and `pos:latitude` are relations. The first four triples describe the fact that `net:host1` and `net:switch1` are connected by `net:link1` through `net:port1` and `net:port2`. The last two describe the location of `net:switch1`, which is at (123.001, 321.001).

Data files formatted by RDF are capable to be queried and reasoned by the computers with semantic query tools, such as SPARQL Protocol and RDF Query Language (SPARQL)[5] [117], [118].

2.3.4 Summary

Summarize from the above, ontology has long been adopted to build knowledge base for knowledge based system. A knowledge based system has been adopted to achieve automation. Knowledge based network management systems have attracted heated discussion in accomplishing network management automation. However, the current research stays as a theory [10], [12]–[15], [119]. The main impediment to achieving practical implementation is the over-complexity and lack of open programmability of the current network systems. Besides, the current telecommunication networks have became increasingly heterogeneous [5]. However, things have changed with the emergence of SDN. The following section will give a detailed introduction of SDN and the benefit it could bring to knowledge based autonomic network management.

2.4 Software Defined Networks

The future of networks is defined by software [4], [120], [121]. The explosion of communication services has resulted in today's network infrastructure becoming increasingly dense and complex [4]. This increasing complexity has led to even more complex network management tasks [4]. Current network management systems spend a significant part of effort on low-level, technology-specific tasks, such as device configurations. With the core idea of separating the network management and physical infrastructure, SDN has been considered as one promising approach to meet those challenges in the future.

[5] https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/

The core concept of SDN is to apply concentrated control over the telecommunication network. The logically centralized control, known as the control plane, works as the brain of the networks, and all the physical infrastructures of communication networks are treated as "dumb pipes". The control plane, working as the network's brain, can obtain, update, and even predict the knowledge base possessing the global knowledge of the network state. It is responsible for the network management, e.g., resource assignment and decision-making. Once a decision is set, it will communicate with the data plane through a particular protocol, e.g., OpenFlow[6], to finish the transmission. OpenFlow is the first SDN standard, and hence the most widely adopted SDN protocol [122]. Its inventors deem it as the enabler of SDN. In the current SDN community, any device that adopts the OpenFlow protocol is deemed as SDN enabled. Not only does OpenFlow configure the network elements, it also provides an open protocol to program the flow-table in different switches and routers. So far, there is no clear definition for SDN, although enormous and increasing number of research papers have been published discussing it [4], [120]–[128]. SDN can be treated as one paradigm, rather than an ossified architecture, where the central software program – the controller, is adopted to optimise and dictate the overall network behaviour. Taking advantage of software engineering and programmability, SDN could solve the problem of network ossification efficiently and make both the control plane and data plane programmable [4], [120]–[122]. It also helps new technologies to be integrated and tested in networks considerably simpler, and therefore accelerates network evolution [4], [120]–[122].

SDN can virtualises the network topology and isolates the traffic of data and control [4], [120]–[122]. The function of the SDN controller is about making decisions on how a connection or a flow is transmitted across the whole network. For example, it can guide end users to select the best accessing network, or provide services from converged networks with multiple technology domains.

[6]Although OpenFlow and SDN are often easily confused, they are two totally different concepts. SDN is a paradigm, while OpenFlow is a protocol defined towards the vision of SDN.

2.4.1 Current Control Strategy of SDN

As stated above, currently there is no clear definition of SDN, thus, any research work that adopts the OpenFlow protocol is deemed as an SDN. The current SDN research mainly took place on different positions and components of the network, as shown in Table 2.2. For example, FlowVisor [123] concentrates on slicing the network physical infrastructure, by placing a slicing layer between the data plane and the control plane, and FlowN [125] is an extension of that work, while RouteFlow [126] focuses on improving the IP routing services. OpenRoads [124] was proposed with the intention to replace current WiFi networks.

Table 2.2: Examples of Current SDN Research.

SDN Applications	**Control Strategy**	**Description**
FlowVisor [123]	Apply software control over any physical hardware, e.g. routers, switches. Place a slice layer between the control and data plane.	Positioned between an OpenFlow switch and several OpenFlow controllers. Can slice any network resources in any control and data plane communication.
OpenRoads [124]	A wireless version of OpenFlow testbed; Adopting FlowVisor, and NOX	Consists three layers: flow layer, slicing layer, and controller layer.
FlowN [125]	An SQL based NOX controller.	Virtualize the network of physical switches.
SoftRAN [127]	An SDN control plane in radio access network	Abstract the base stations network into a virtual big base station.
OpenRAN [128]	Provides “match-action” control strategy	Consists of three parts: wireless spectrum resource pool, cloud computing resource pool, and SDN controller. It provides abstraction on the resources in both data and control plane.

There are two main components in RouteFlow, an OpenFlow controller and a RouteFlow server running as an application on top of it. There is a virtual network topology of virtual machines, with the routing engine installed on each of them [126]. The virtual topology mirrors the physical topology, but may not necessarily match the exact topology of the real physical infrastructure. That is to say, with different real-time communication services requirements, the controller will mirror the real physical network in different ways. There are three main mapping schemas: $1:1$, $1:n$ and $m:1$ or $m:n$, in which m and n are positive integers, indicating the numbers of nodes

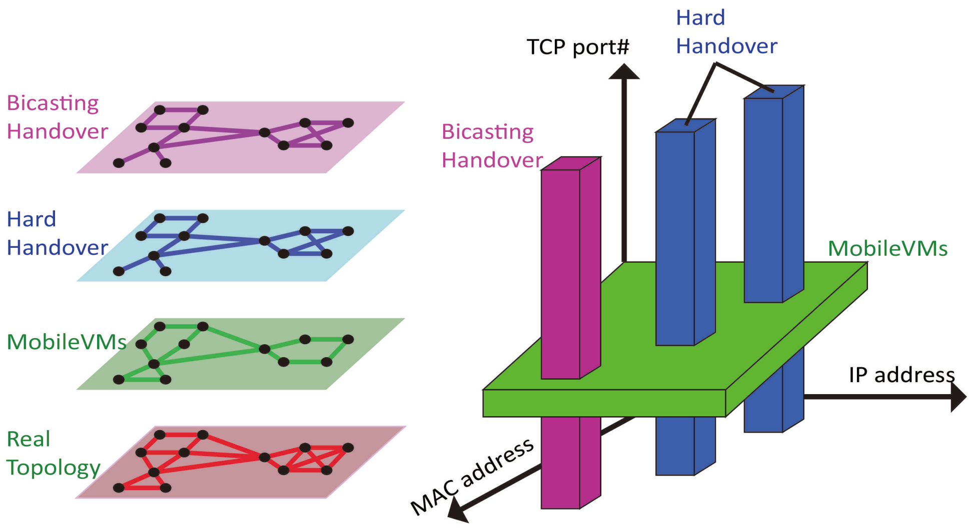


FIGURE 2.2: Demonstration of Network Slicing in FlowVisor [2].

in physical network topology and logical network topology. These three mapping schemas represent logical split, multiplexing, and aggregation, respectively [126].

FlowVisor is an SDN controller built on OpenFlow [2], [123]. It has already been implemented on a small network at Stanford University since May 2009. FlowVisor acts as a transparent proxy between OpenFlow switches and other controllers supporting OpenFlow, such as NOX. In FlowVisor, network resources are sliced in various dimensions, e.g., by bandwidth, topology or device CPU, through a transparent slicing layer. It provides a strong separation between the control and data planes, by slicing the network resources into many orthogonal and independent slices [127]. It can enforce block and rewrite control messages as they pass through [127]. A demo of network slicing with FlowVisor is given, in which four isolated slices over the same physical network are demonstrated, as shown in Fig. 2.2. In these slices two of them are wireless and two are wired [2], [123]. FlowN adopts a similar strategy with FlowVisor, except that it adopts a database to help with the mapping between the control plane and data plane [125]. The advantages of FlowN over FlowVisor fall into two categories: 1). FlowVisor does not supply a full virtualization over the physical resources as FlowN does. It simply slices them. 2). the mapping algorithm FlowVisor adopts is not efficient. For example, it has to iterate every node when doing a mapping [2], [125]. In FlowN, mapping can be done by a simple SQL query. It has been proven to be more scalable in a large network that may contain more than 100 virtual networks [125]. However, FlowN with memcached is still slower than FlowVisor. As a summary, adopting database technology in virtual network mapping is a promising trend, but a lot of work is still required.

The OpenFlow protocol could be applied in different domains and different layers of the network. For example, RouteFlow [126] tries to execute centra-dictatory IP routing on computer networks; while SoftRAN and OpenRAN [127], [128] concentrated on providing software defined centralised control on radio access network. Odin, a SDN framework based on WLAN, tries to abstraction virtual wireless access points from real physical switches [129]. OpenRoads is a SDN application based on FlowVisor [125]. Both FlowN and OpenRoads implement the OpenFlow protocol over physical switches.

2.5 Knowledge Based Autonomic Network Management in SDN

The potential of open programmability and logically centralized knowledge control in SDN paradigm offer an attractive approach for a plurality of technologies and applications [130]–[132]. Accomplishing a knowledge based autonomic network management system is one of them [12], [13], [15], [20], [65], [110], [133]–[136]. Shared semantic annotations, not only enable data integration from heterogeneous data resources, but also helps to format newly generated data into historical knowledge base.

SDN-based architecture provides a centralised controller for the network with the global knowledge of the network which, thus, is capable of controlling the network infrastructure in a vendor-independent manner. The network devices simply just accept policies given from the controller without understanding the full state of the network. They simply implement various network protocol standards, resulting in

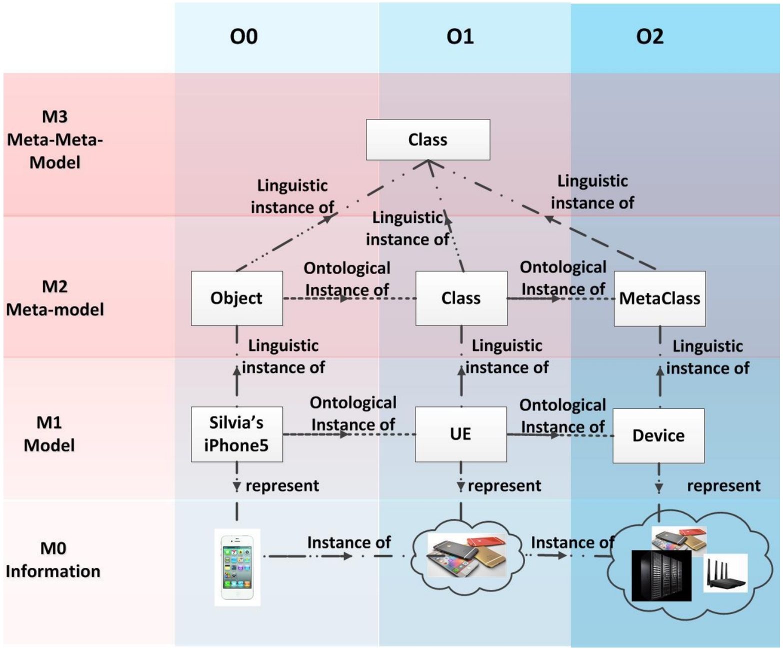


FIGURE 2.3: An example of linguistic meta-modeling and ontological meta-modeling for SDN and their relationship.

directly control, program, orchestrate, and manage network resources at the SDN controller, and saving a lot of workforce and resources.

2.5.1 Current Knowledge Based Autonomic Management

In the semantic knowledge based network management system [12], [14], [21], [58], [62], [133]–[137], [137]–[143], [143]–[146], a resource can be any component in the network, e.g., a port, a router, or a mobile phone. The ontology is adopted to construct these resources into a vast, machine-readable database. Then, artificial intelligence approaches, e.g., machine learning and data mining can be adopted to discover the information hidden in the database.

2.5.2 Semantic Meta-Modelling Architecture

The process of network knowledge base building is typically a network modelling process, which in its heart is an object-oriented software development process. From abstract to concrete, different models of the network can be developed. To illustrate the connections of models with different levels of abstraction, concepts of meta-modelling are borrowed from software engineering to describe the knowledge base developing process for communication networks. The network modelling process with different levels of abstraction is shown in Fig. 2.3. By adopting the ontology to describe network resources, the modelling can be distinguished into linguistic meta-modelling and ontological meta-modelling [147]–[149].

There are four layers in the linguistic meta-modelling hierarchy, M0 is the information layer, M1 is the Model Layer, M2 is the Meta-Model Layer, and M3 is the Meta-Meta-Model layer. A model is a representation of a particular domain, which is constructed and manipulated for a specific purpose [147]–[149]. The meaning inherent in a model depends on the circumstances where it is used. It is just a syntactic representation of the information. With proper interpretation, the information can be extracted and understood from the data [148], [149].

A Meta-model is a model that describes modelling languages. A model is an instance of the meta-model, as shown in the vertical "instance of" relations in Fig. 2.3. M2 and M3 are the language-specification layers, while M1 is the user specification layer. M3 contains only one meta-model – Meta Object Facility (MOF, a core technology in model driven engineering, which is a new trend of software engineering) [148], and M2 contains meta-models that can specify the models represented in M1 layer. M1 defines all the data. M0 is the run-time layer, it represents the objects instantiated from the models. In computer science, it represents runtime systems. In communications, however, it represents real-time communication physical network infrastructure. In the ontological meta-modelling dimension, it is divided into three layers, O0, O1, and O2. It describes the user equipment (UE) resource for the proposed knowledge base.

The ontological meta-modelling is used to define domain specific properties. There are two different types of "instance-of" relation [147]. The elements in M1 layer are linguistic instances of those in M2 layer, and the elements in O0 layer are ontological instances of those in O1 layer. By dividing the meta-model into linguistic and ontological dimensions, the model can provide a description more grounded in real wireless networks [147]. The relation and distinction between syntax and semantics is

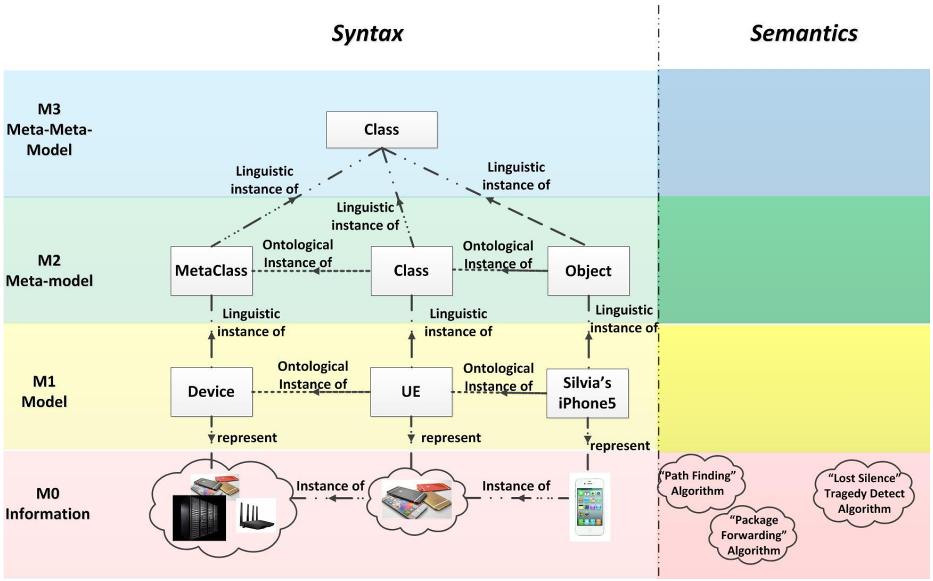


FIGURE 2.4: An example of syntax and semantic domain of wireless network knowledge base and their mappings.

illustrated in Fig. 2.4. It gives an example of syntax and semantic domains of wireless network knowledge bases and their mapping. The value of a model is fully achieved only when the modelling concepts directly map with domain concepts rather than computer technology concepts. Thus, when building on knowledge base for communication networks, the model concepts need to be directly mapped to network domain concepts. In the semantic domain, the abstract syntax is related to well-defined and well-understood concepts in SDN, e.g., Shannon's Law, path finding algorithm, and packet forwarding algorithms.

2.5.3 Steps of Knowledge Base Building from Syntax to Semantic

Creating a knowledge base takes two steps, as shown in Fig. 2.5 [8]. The elements in the physical infrastructure of a network, such as ports, switches, routers, cables, etc., are called network elements [8]. The first step is to map the network elements

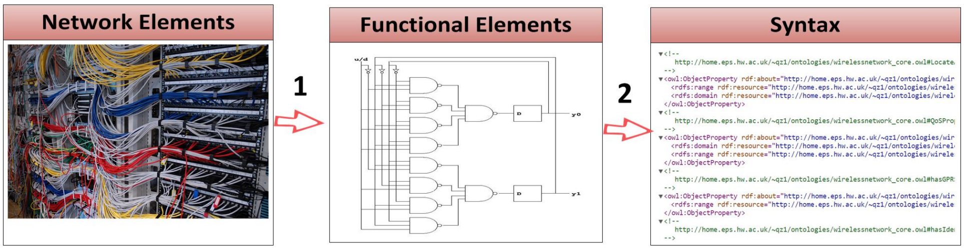


FIGURE 2.5: Steps to build a network information model [5].

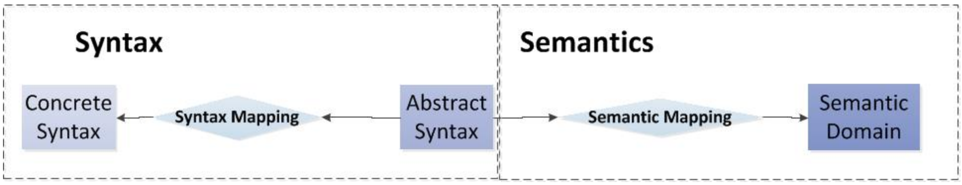


FIGURE 2.6: Syntax and semantics of a modeling language [58].

to functional elements. Then, the RDF and Web Ontology Language (OWL), are adopted to map all the functional elements into syntax.

The relationship between syntax and semantics is illustrated in Fig. 2.6. Many existing applications for the network descriptions are in syntax, such as MIB [46], NDL [16], and RDF [150]. Based on the syntactical description, some semantic description technologies have been developed. A widely adopted approach is OWL, a metadata standard developed by W3C[7] [96]. As an expressive ontology language, OWL separates the ontology from syntax. The ontology provides an explicit representation of semantics, the separation of the ontology and the syntax allow users to mix different ontologies without being concerned about the syntax with which they are described [151]. Besides, valuable information can be extracted from the technology-independent resource by data mining.

2.5.4 A Schematic of Semantic Knowledge Base Building Process

A schematic of semantic knowledge base developing process for SDN is illustrated in Fig. 2.7, on the basis of a widely-accepted 5G cellular architecture [3]. The key

[7] https://www.w3.org/OWL/

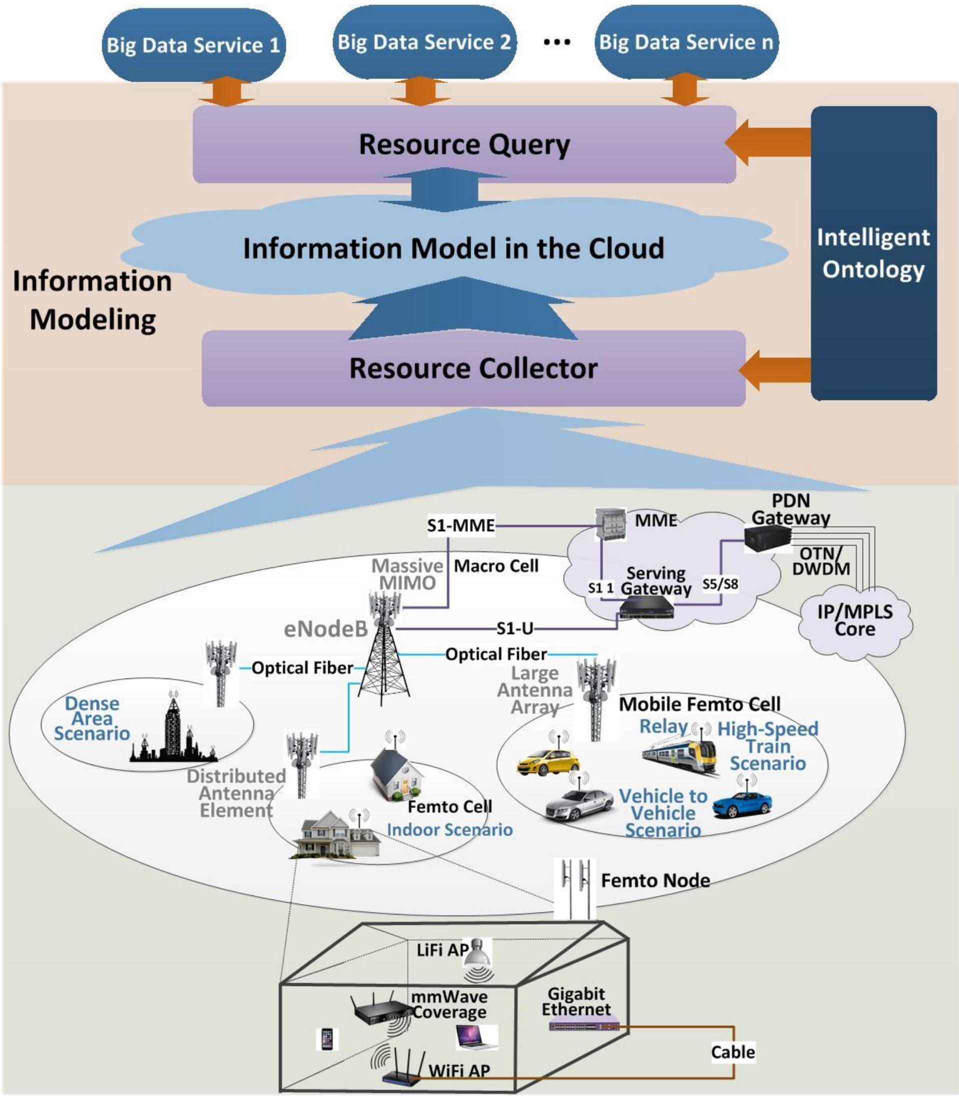


FIGURE 2.7: A schema of information modelling in hybrid telecommunication networks with various technologies, based on a widely accepted future 5G wireless network stereotype [3].

component is the SDN resource description in Fig. 2.7. The resource collector can describe the resources of the physical infrastructure in the knowledge base, which is stored in cloud-based servers. Thus, the knowledge base can be abstracted without losing any necessary technology details. Then, a query language such as SPARQL[8] can be adopted to search and edit the knowledge base [117]. The way to apply universal semantic resource description in SDN is by exchanging RDF data between SDN operation systems. The ontology works as a language between different SDN operation systems.

[8] https://www.w3.org/TR/rdf-sparql-query/

2.6 Software Tools

To develop a semantic knowledge based network management system, software tools for both the semantic web technologies and SDN are required. The current software tools are introduced below.

2.6.1 Software Tools for Knowledge Engineering

- ***Protégé***[9]***:*** Protégé is an open-source, ontology editor providing a GUI. It provides a GUI editing viewer to edit ontologies and instances [81].

- ***Apache Jena***[10]***:*** Jena is a JAVA framework to develop semantic web applications. Jena provides APIs to generate and read RDF or OWL triples. Like Protégé, Jena also has some internal reasoners. It also supports SPARQL to query the knowledge base. It consists of the following components, RDF API, ontology API, inference API, and ARQ (a SPARQL engine) [152].

- ***RDFlib***[11]***:*** RDFlib is a light-weight yet powerful Python library, providing the following functions, ontology editing, namespace managing, RDF triples iterating, and SPARQL querying [153].

- ***Stardog***[12]***:*** Stardog is a commercial triplestore. Combined with graph database Apache Tinkerpop[13], Stardog could generate a RDF database with graph database features as well. For example, a stardog database could find the shortest path between two nodes, which is not yet supported by RDF knowledge graph and SPARQL [154].

[9] https://protege.stanford.edu
[10] http://jena.apache.org
[11] http://rdflib.readthedocs.io/en/stable/
[12] https://www.stardog.com
[13] https://tinkerpop.apache.org

2.6.2 Software Tools for SDN

- *Mininet*[14]*:* it is an SDN emulator, creating realistic virtual networks with simple commands. Mininet runs on top of a virtual machine (usually VirtualBox) or on a cloud. It supports the OpenFlow protocol and Open vSwitch (OVS) tools to develop SDN applications [155].

- *OVS*[15]*:* targeted at highly dynamic, multi-server virtualization environment, OVS is a multi-layer virtual switch [156]. By trying to reuse the code as much as possible, OVS provides a plethora of powerful APIs for autonomic network control, and thus provides a high abstraction of the dynamic networks. It is worth mentioning that, for the Linux systems later than 3.3, OVS is provided as a part of the kernel, which makes application developing with OVS much more convenient.

2.7 Research Gap

Knowledge engineering has gained more and more attention in the domain of network management [12], [14], [21], [58], [62], [133], [137], [137]–[143], [143]–[146]. Here we only focus on those related to semantic web technologies. For example, the ontological knowledge based network management systems proposed by J. E. López de Vergara, et al. [12], [138]–[143], John Strassner, et al., [14], [137], [143]–[146], Ohshima, R. et al, et al [21], are discussed and assessed.

J. E. López de Vergara, et al. [12], [138]–[143], mentioned the possibility of adopting an ontology to map and merge the current heterogenous knowledge bases, and provide a universal view of the whole managed system.

Strassner et al [14], [137], [143]–[146] investigated the possibility of ontology-driven policy based network automatic management system. They proposed a series of autonomic network management systems:

[14]http://mininet.org
[15]http://www.openvswitch.org

- *Directory Enabled Networks next generation (DEN-ng) policy model* [137]: A semantic intelligent policy model, enable semantic policy reasoning with ontologies and machine learning.

- *Policy Continuum* [157]: Translating high abstraction level policies to low level policies automatically, hiding the complex configuration tasks. they primarily focused on theoretical architectures.

- *Context-aware policy model* [158]: Adopting ontology to represent context of the existing network management data, enabling semantic reasoning on the policies.

However, as described in the above reports, with the absence of practical approaches, these systems have not been evaluated in real life scenarios.

In [21], Ohshima, R. et al proposed a network management system with the assistance of an ontology, collecting configuration information from traditional network management databases. However, the work was based on a traditional network and did not provide technology-independent, reusable APIs.

Zied Ben Houidi [6], explored the practicality of knowledge based autonomic reasoner for network management. He implemented the proposed method in python, using Pyke as the reasoner, and tested the system with the basic network connectivity problem. However, in this work, some complexities are hidden for the purposes of clarity. Besides, it is still not clear whether the system work in other more dynamic and complicated situations.

Debao Xiao et al. [15], tried to integrate ontology with policy based approaches to accomplish autonomic network management. They also proposed a possible scenario. However, there is no practical tools ready to implement for the proposed methodology yet, or any real life evaluations for the proposed system.

In summary, there is still a gap between the current knowledge based autonomic network management research and their implementations in real life use cases. The current research still focuses on theoretic framework and architectures.

2.8 Conclusions

In this chapter, we have produced an overview of the research topics of SDN and knowledge engineering. It shows that the emergence of SDN has provided a promising approach to bring autonomic network management into practise. As a key technology of Artificial Intelligence, knowledge engineering is an attractive approach to accomplish autonomic network management. The software tools for SDN and knowledge engineering are also presented.

The next chapter will present the detail of our approach in developing a knowledge base for the SDN management.

Chapter 3

An Ontological Model for Software Defined Networks

3.1 Introduction

As stated in 2.3.3.3, the ontologies proposed for network management are numerous. However, they are designed for specific tasks. There is no single "best" approach for the domain of network management. They are not yet able to provide a universally accepted knowledge base for telecommunication networks with hybrid technologies. There are three main reasons for this:

- First, many network description ontologies are proposed for particular applications, rather than for the overall network resources.

- Second, the evaluation of ontology is problematic. Although many evaluation theories have been put forward [85], [86], few reports detail how to carry out the evaluation step-by-step. Generally speaking, for network description ontologies, there are two approaches to evaluation. One is to discuss with experts in the specific field, the other is to apply it in a real-world application. To the best of our knowledge, very few use cases have been carried out in practice. Thus, it is difficult to determine that any particular ontology is superior to any other.

- The final reason lies in the ever-changing nature of communication technology. For example, wireless communication technology changes generation almost every decade. New technologies keep arising, and it is difficult to develop a standard of the vocabulary to describe them.

The detail of the development process for our proposed ontology, ToCo, is given in this chapter. The ToCo ontology, which has been published according to best practice[1], is constructed into six modules: namely, `Device`, `Interface`, `Link`, `User`, `Service`, `Data`, as shown in Fig. 3.1 (source file of the ontology is given in Appendix B). The entire ontology consists of 37 concepts, 20 object properties, and 29 datatype properties. ToCo is able to describe the physical infrastructure of hybrid telecommunication networks, including devices, interfaces, and links in both wired and wireless networks. The quality of a communication channel can also be described, via bandwidth, data rate, package loss, delay, etc., to give a detailed description of the performance of the network. Finally, details of the services provided by telecommunication networks, as well as customers, are also included, as they are an important aspect of telecommunication systems. Our domain of interest is a network of networks with heterogenous technologies. There is a universal knowledge graph for all the networks. This knowledge graph suits LiFi, WiFi, Millimeter Network, LTE-Massive MIMO, optical backbone network, etc. Information will be abstracted from each of these domains, and all these knowledge graphs will be constructed into one knowledge base. However, it is not the case that sub-ontologies should be created for each of these technical domains, instead, there will be one unique ontology infrastructure describing all of the domains.

The major contributions of this chapter are summarised as follows.

- The main contribution is the ontology ToCo. An outline and the key modules of ToCo are presented in Section 3.2.

- The second contribution is the Device-Interface-Link (DIL) pattern based on which ToCo is built. It can describe the relationship in any kinds of networks, or more generally, linked things, as presented in detail in Section 3.2.

[1] http://purl.org/toco/

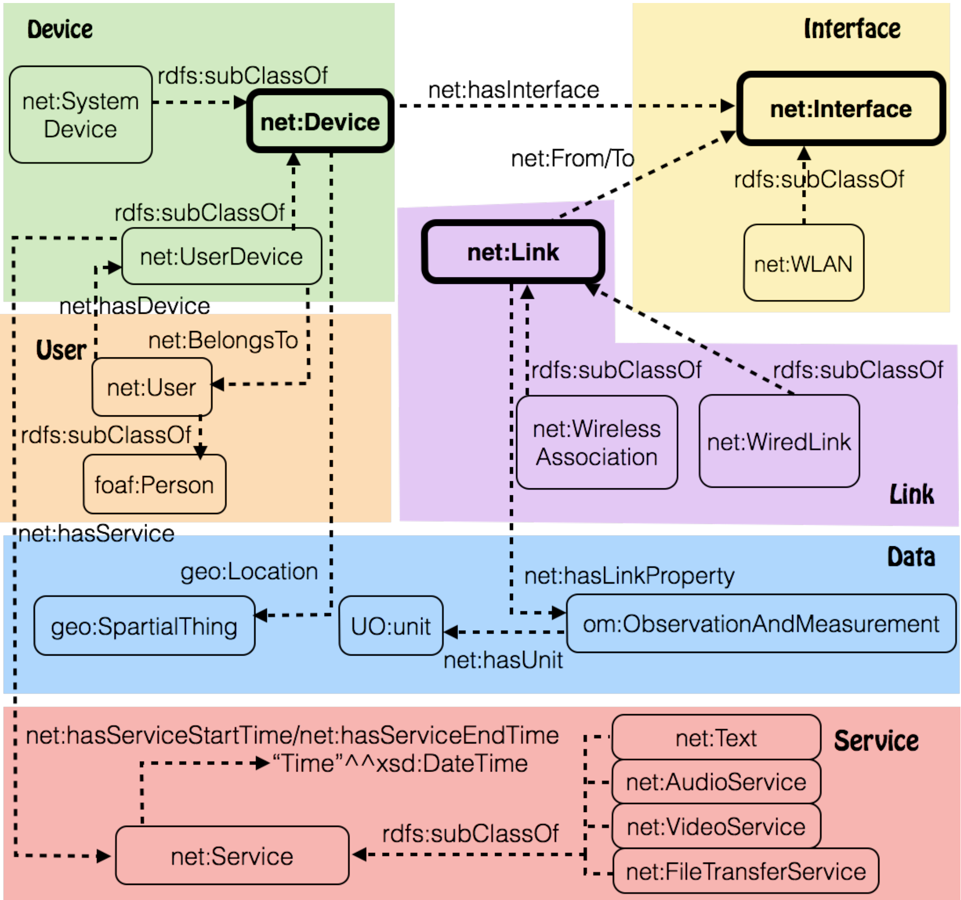


FIGURE 3.1: The main concepts and properties of ToCo ontology. The concepts are in boxes, while the key concepts (that make up the DIL pattern) are in bold boxes. The main properties are illustrated in dashed lines. ToCo can be seen from six perspectives: namely, `Device`, `User`, `Interface`, `Link`, `Data`, and `Service`. These perspectives are denoted with different colours. The central concepts are brought out by the `DIL` pattern, composed with the main perspectives `Device`, `Interface`, and `Link`, which is able to abstract the knowledge pattern of all kinds of networks.

- The third contribution are the examples of ToCo, describing networks with various technologies, and the use cases in which ToCo is used (Section 3.3). This is followed by a conclusion (Section 3.4).

We begin the developing process by reviewing existing ontologies and telecommunication standards. As the telecommunication networks become increasingly dense, heterogeneous and complex, it is necessary to read the multiple recommendations, technical specifications, protocols, and white papers to obtain the knowledge of the telecommunication systems.

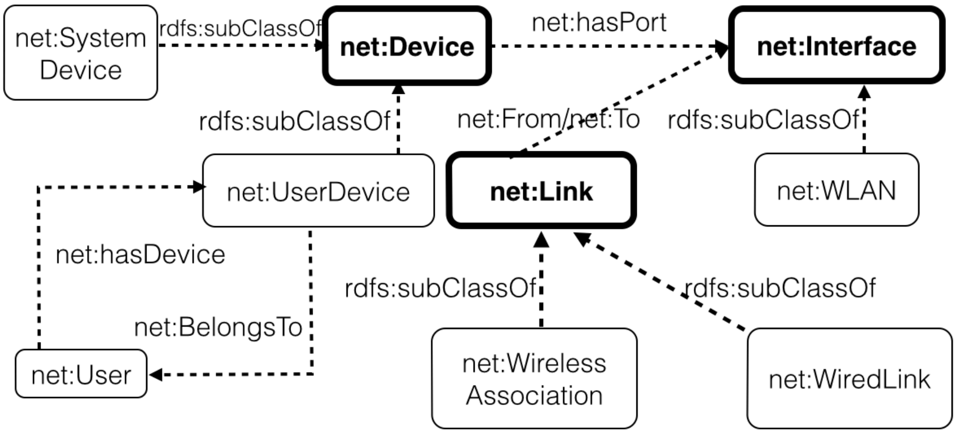


FIGURE 3.2: The Device-Interface-Link Pattern (central concepts in bold).

3.2 TOUCAN Ontology

As is generally known, ontology engineering is at its heart a modelling endeavour [74]. During the modelling process, networks with different access technologies are observed to have repeating structurally similar knowledge patterns, as shown in Fig. 3.2. No matter what kind of network we are modelling, essentially they are all devices with interfaces through which we can connect. Although they have different devices (e.g., WiFi Access Point for WiFi network, Base Station towers for LTE networks), and different link (e.g., the frequency of the radio transmitted in WiFi, LTE, and LiFi are different, and Computer network use twisted cable), and the technology protocols adopted in different networks are also different, but in the core, these networks can all be deemed as linked devices. This pattern describes the basic relations of devices, interfaces, and links, I named it the `DIL` pattern. ToCo is built around that pattern. The `DIL` pattern identifies and provides an important insight into the abstract and recurring knowledge structure in the domain of networks, or more generally, any kinds of linked things [74], [159]. With the `DIL` pattern, the ontology modelling processes for networks are made clearer and efficient. ToCo holds an inclusive view of the telecommunication networks: "devices with interfaces through which you can connect". Thus, from small-scale telecommunication networks such as vehicle-to-vehicle networks [160], [161], smart home devices [162], [163], to large-scale networks such as satellite networks [164], [165] can all be described with ToCo. The source code of ToCo is available at Appendix B.

ToCo can be seen from six perspectives:

- A device perspective: focus on the devices in the network and their properties;
- An interface perspective: focus on the interfaces on the devices, and their properties;
- A link perspective: focus on a link, wired or wireless, between two interfaces, and its properties;
- A user perspective: focus on the users of user equipments, their information and properties.
- A data perspective: focus on the data measured or observed out of a property.
- A service perspective: focus on the service provided by telecommunication system for users.

The following sections describes the `DIL` pattern and the `device, interface, link, user, data and service` perspectives. The namespaces used are written as the prefixes shown in Table 3.1.

TABLE 3.1: Prefixes and namespaces used in the ToCo ontology.

Prefix	Namespace
net	`<http://purl.org/toco/>`
xsd	`<http://www.w3.org/2001/XMLSchema#>`
geo	`<http://www.w3.org/2003/01/geo/>`
foaf	`<http://xmlns.com/foaf/spec/>`
om	`<http://purl.oclc.org/net/unis/ontology/sensordata.owl/>`
UO	`<http://purl.obolibrary.org/obo/uo.owl>`
rdf	`<http://www.w3.org/1999/02/22-rdf-syntax-ns#>`
rdfs	`<http://www.w3.org/2000/01/rdf-schema#>`

3.2.1 Device

A device (`net:Device`) is the device in the physical infrastructure of the telecommunication networks, with the ability of transmitting and/or receiving signals (based on

the frequency, could be microwave, million-meter wave, optical wave, etc.). Based on the function and role played in the telecommunication networks, devices can be divided into system device (`net:SystemDevice`) and user device (`net:UserDevice`). Moreover, the devices in networks of a specific technology domain are subclasses of (`rdf:subClassOf`) the device (`net:Device`), for example, in wired network, there are hosts (`net:Host`) and switches (`net:Switch`); in LTE network, there are base stations (`net:BaseStation`) and user equipment (`net:UserEquipment`); in WiFi and LiFi networks, there are access point
(`net:AccessPoint`), which can be further divided into WiFi access point
(`net:WiFiAccessPoint`) and LiFi access point (`net:LiFiAccessPoint`). The ontology view of Device is shown in Fig. 3.3.

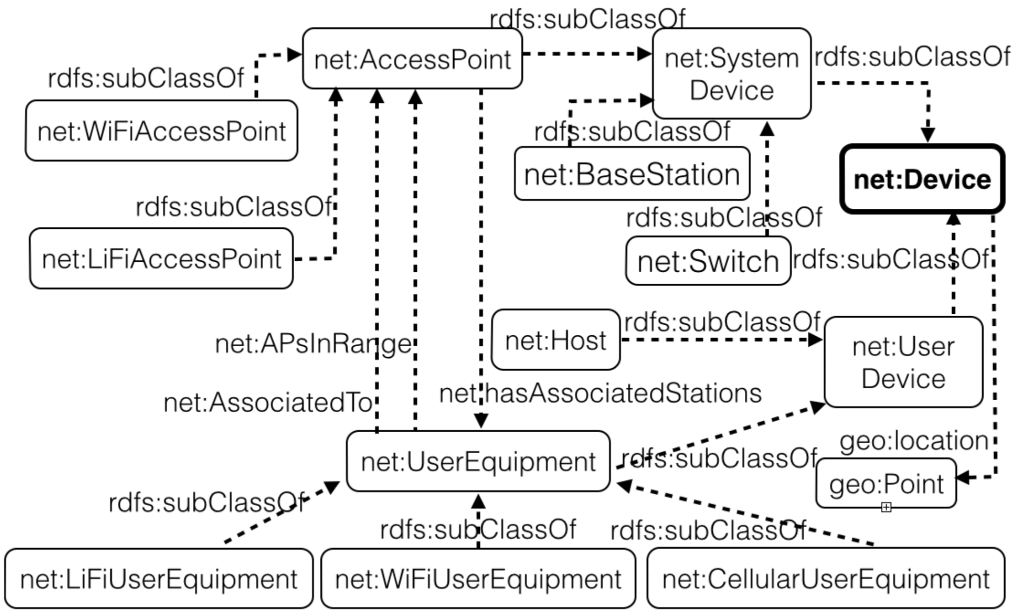


FIGURE 3.3: Ontology view focusing on Devices.

3.2.2 Link

A link (`net:Link`) is one of the most important concepts in telecommunication networks. The principal obligation of the telecommunication network is to establish a link and improve the quality of the link. A link could be a wired cable (`net:WiredLink`), or a cluster of wireless electromagnetic wave (`net:WirelessAssociation`). Please be noted that `net:WiredLink` and `net:WirelessAssociation` are distinct with each other, e.g., a link cannot be both at the same time.

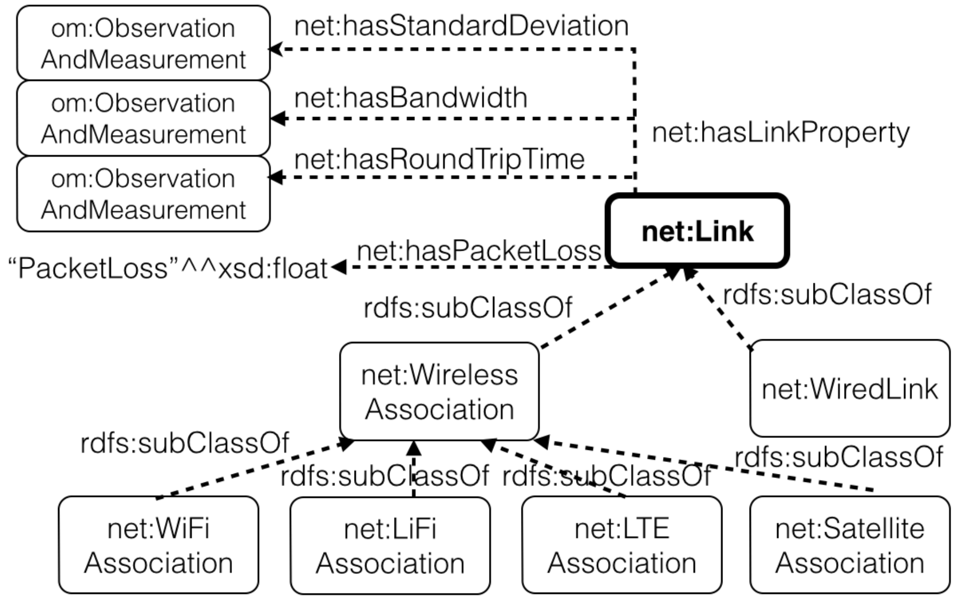


FIGURE 3.4: Ontology view focusing on Links.

The properties of links determine the quality of a communication, for example, bandwidth (`net:hasBandwidth`), data rate (`net:hasDatarate`), transmit power (`net:hasTxpower`), receive power (`net:hasRecpower`), etc. An example of describing the bandwidth of a `Link` is shown below.

```
net:link_1 a net:Link ;
 net:hasBandwidth net:link_1_bw .
net:link_1_bw a om:Measurement ;
 net:hasValue "50"^^xsd:float ;
 net:hasUnit UO:0000325 .
```

It describes the fact that a link `link_1` has a bandwidth of $50MHz$. The ontology view of `Link` is shown in Fig. 3.4.

3.2.3 Interface

Interface (`net:Interface`) describes the wired (`net:Port`) and wireless interfaces (`net:WLAN`) on the devices in telecommunication networks. Most of the important information for network routing is described as the properties of the interface, for example: IP address (`net:hasIP`), MAC address (`net:hasMAC`), antenna gain (`net:hasAntennaGain`, a property for wireless interface `net:WLAN`), etc. The ontology view of `Interface` is shown in Fig. 3.5.

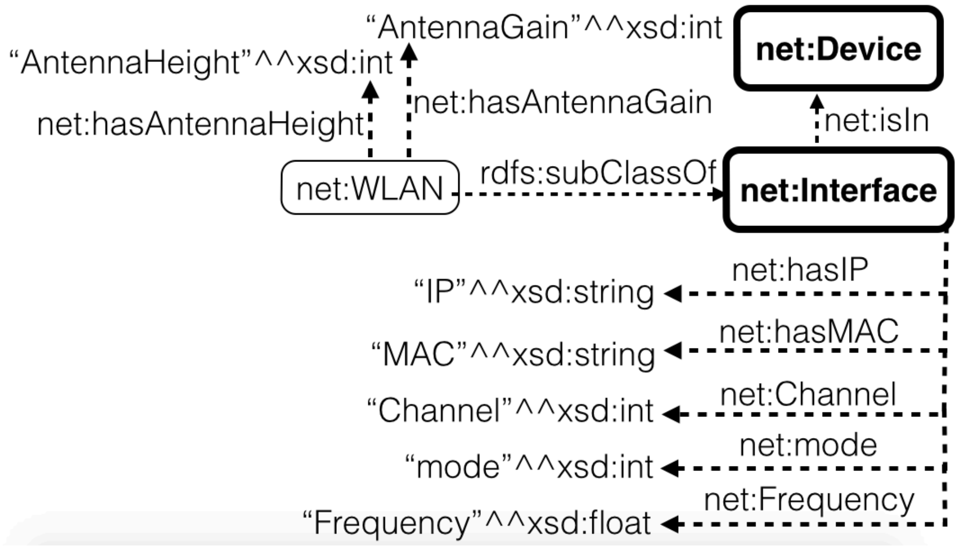


FIGURE 3.5: Ontology view focusing on Interface.

3.2.4 User

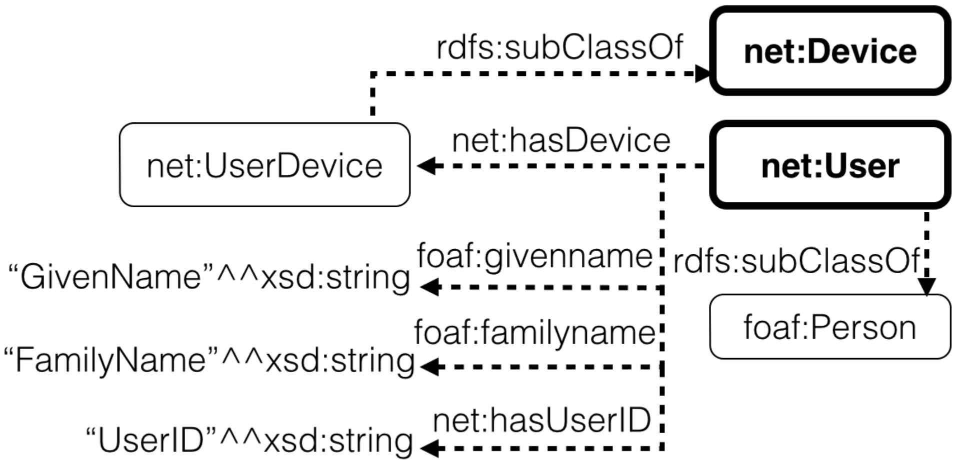


FIGURE 3.6: Ontology view focusing on User.

The user information in telecommunication networks includes `user id`, `family name`, `first name`, `home country`, `home town`, etc. As the user is a human in real life, parts of the `foaf` ontology[2] is reused. The main relationship between `User` is with the `net:UserDevice`:

`net:User net:hasDevice net:UserDevice.`

Some main concepts of `User` are shown in Fig. 3.6.

3.2.5 Data

All the observation and measurement data, location and time information are described in the data module. These general information, such as location, time, measurement, have already been modelled by ontologies. The popular ontologies are reused here to describe the data. For example, the Units Ontology (UO)[3] is reused to describe the units of the data [166]. The SENSEI[4] observation and measurement ontology[5] is reused here to describe the observation results and measured data in telecommunication system. Location information are described with WGS84 ontology[6]. We also reused time ontology[7] to describe time information.

3.2.6 Service

The service module describes the details of telecommunication services, e.g., voice session, video session, document transmission. Some concepts of the service module are shown in Fig. 3.7.

[2] http://xmlns.com/foaf/0.1/
[3] http://purl.obolibrary.org/obo/uo.owl
[4] A project of EU, http://www.sensei-project.eu/
[5] http://purl.oclc.org/net/unis/ontology/sensordata.owl
[6] http://www.w3.org/2003/01/geo/wgs84_pos/
[7] http://www.w3.org/2006/time#

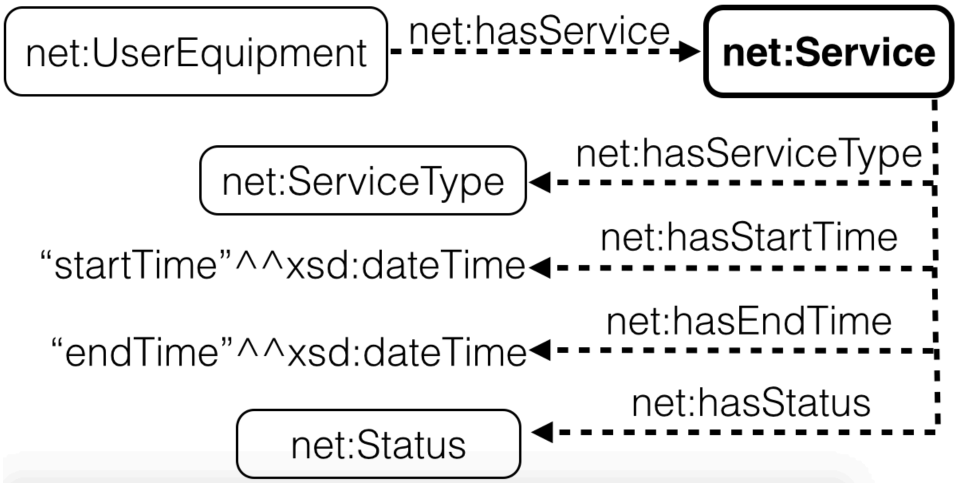


FIGURE 3.7: Ontology view focusing on Service.

3.3 Examples and use cases of ToCo

3.3.1 Examples

Examples are provided to demonstrate how networks within different technology domains are described with ToCo. The examples include: three network resource description examples for WiFi, LiFi, and computer network, respectively, and a SDN flow description example.

To describe the device information, a simplified schema of a WiFi network is shown in Fig. 3.8. The x-axis and y-axis denote the longitude and latitude of a planar graph. The circles with different colours represent WiFi access points (circles in blue) and user equipments like phones, laptops (circles in red), with the area of circles denotes the cover range of signal. If the centre of a red circle is in range of the blue circle, it means this user equipment is in the range of the WiFi access point. Some of the main triples are shown in Listing 3.1. As shown in the Listing, the information of access point, such as, the half intensity angle, optical transmitted power, mobile stations in range, and the location is represented without redundancy.

The Fig. 3.9 shows a three-dimensional coordinate where a LiFi access point "LiFi1" and a user device "sta1" are located. The information of access point, such as, the half

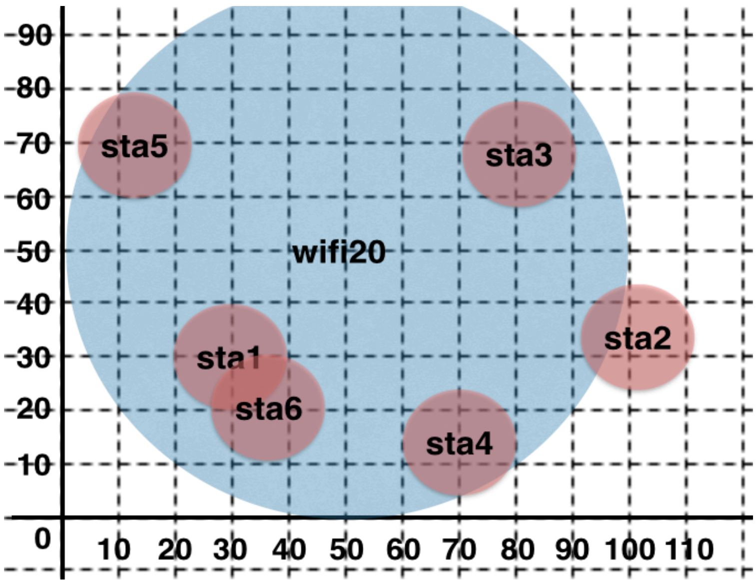


FIGURE 3.8: The schema of a WiFi network with 1 access point "`wifi20`" and six mobile stations "`sta1`" to "`sta6`".

intensity angle, optical transmitted power, mobile stations in range, and the location is represented in Listing 3.2, as well as the information of the wireless link between "LiFi1" and "sta1", such as distance, bandwidth, incident angle, and radiance angle.

```
net:wifi20 a net:WiFiAccessPoint ;
    net:driver "nl80211"^^xsd:string ;
    net:hasWLAN net:wifi20-wlan1 ;
    net:ssid "wifi"^^xsd:string ;
    net:stationsInRange
        net:sta1, net:sta2, net:sta3 ,
        net:sta4, net:sta5, net:sta6 ;
    net:associatedStations net:sta1 ;
    geo:location net:wifi20_location .
net:wifi20_location geo:alt  "0.0"^^xsd:float ;
    geo:lat "50.0"^^xsd:float ;
    geo:long "50.0"^^xsd:float .

net:sta1 a net:WiFiUserEquipment ;
    net:hasWLAN net:sta1-wlan0 ;
    net:hasName "sta1"^^xsd:string ;
    geo:location geo:sta1_location .

net:sta1_h1 a net:WirelessAssociation ;
    net:from net:sta1 ; net:to net:h1 ;
    net:hasBandWidth net:sta1_h1_bw .
net:h1_sta1_bw a om:Measurement ; net:hasUnit UO:0000325 ; net:
   hasValue "51.5"^^xsd:float .
```

LISTING 3.1: Part of the RDF knowledge graph for a WiFi network. The knowledge of the stations in range, stations associated, location of a WiFi access point, and the bandwidth of the wireless link between the WiFi access point and a user device "sta1" is described.

To describe a wired computer network, some examples of the triples are shown in Listing 3.3. Knowledge about the ports, e.g., IP address, MAC address, and link

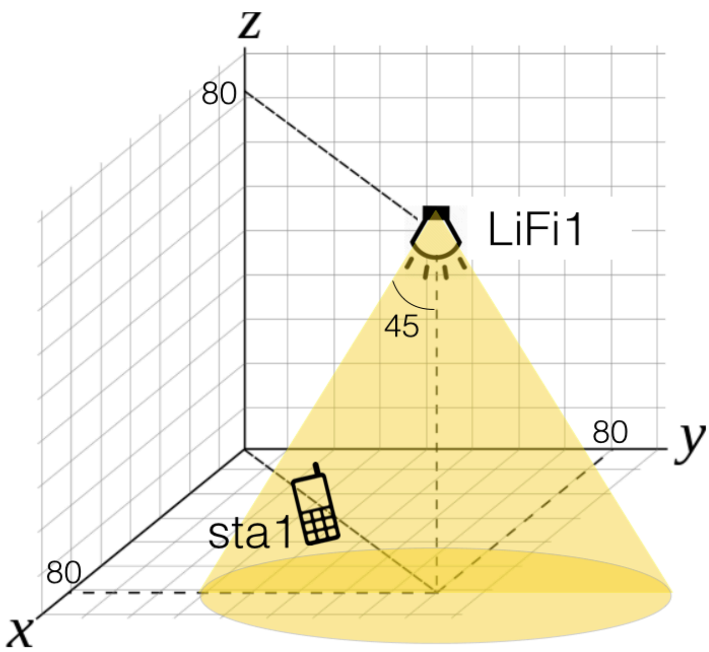


FIGURE 3.9: A schema of a LiFi network with one access point and one mobile station.

information such as bandwidth are described.

```
net:LiFi1 a net:LiFiAccessPoint ;
    net:hasGainOfOpticalFilter "1"^^xsd:int ;
    net:hasHalfIntensityAngle  "45.0"^^xsd:float ;
    net:hasOpticalTransmittedPower "0.3"^^xsd:float ;
net:hasRespansivity "1"^^xsd:float ;
net:stationsInRange net:sta1 ;
geo:location net:LiFi1_location .

net:sta1 a net:LiFiUserEquipment ;
    net:hasFieldOfView "90"^^xsd:float ;
    net:hasGainOfConcentrator  "1"^^xsd:float ;
    net:hasWLAN net:sta1_wlan0 ;
geo:locate geo:sta1_location .

net:sta1_wlan0 a net:WLAN ;
net:hasAssociation net:sta1_ap1 .

net:sta1_ap1 a net:LiFiAssociation ;
    net:hasDistance "9"^^xsd:float ;
    net:hasIncidentAngle "15"^^xsd:float ;
    net:hasRadianceAngle "27.5"^^xsd:float ;
    net:hasBandwidth net:sta1_ap1_bw .
net:sta1_ap1_bw a om:Measurement ;
    om:hasValue "5"^^xsd:float ;
    om:hasUnit UO:0000325 .
```

LISTING 3.2: Part of the RDF knowledge graph for a LiFi network. The knowledge about the half intensity angle, optical transmitted power, mobile stations in range, and location of a LiFi access point is represented, as well as the knowledge of the association between “LiFi1” and “sta1”, such as distance, bandwidth, incident angle, and radiance angle.

```
net:s1 a net:Swtich ;
net:hasPort net:s1_eth0, net:s1_eth1, net:s1_eth2 .
```

```
net:h1 a net:Host ;
net:hasPort net:h1-eth0, net:h1-eth1, net:h1-eth2 .

net:h1-eth0 a net:Port ;
    net:hasIP "10.0.0.1" ;
    net:hasMAC "f6:8a:d8:0b:6d:e7" ;
net:isIn net:h1 .

net:s1_eth1 net:hasLink net:s1_h1 .
net:s1_h1 a net:wiredLink ;
    net:hasBandwidth net:s1_h1_bw .
net:s1_h1_bw a om:Measurement ;
    net:hasValue "50"^^xsd:float ;
    net:hasUnit UO:0000325 .
```

Listing 3.3: Part of the RDF knowledge graph of a computer network, describing the information about the ports and links, e.g., IP address, MAC address, and bandwidth.

Another example is about the SDN. The SDNs are about making decisions on how a flow (or a connection) is transmitted across the whole network. Thus, `Flow` is the key concept in SDN. ToCo is able to describe the properties of the `net:Flow`, as shown in Listing 3.4.

```
net:s1 net:hasFlow net:s1_flow1 .
net:s1_flow1 a net:Flow ;
          net:idle_timeout 0 ;
          net:table_id 0 ;
          net:flags 0 ;
          net:hard_timeout 0 ;
          net:priority 0 ;
          net:cookie 2 ;
          net:hasAction
              net:s1_flow2_action0 .

net:s1_flow2_action0 a
        net:Output ;
        net:toPort net:s1_port1 .
```

Listing 3.4: Part of the knowledge graph of a Flow in SDN. The information of a Flow is mainly described.

3.3.2 Use cases

ToCo has been used in several applications for autonomic network management and disaster response. These use cases are: a network autonomic management system "SEANET", a network policy-based management application, a shipwreck early detection use case "lost silence" [39], and "SARA", a resource allocation application in post-tragedy situation.

SEANET is a technology independent, knowledge-based network management system. The ToCo ontology and the `DIL` pattern are the key for the SEANET. It adopts the

ToCo ontology as the language to build the knowledge base for telecommunication networks, and use SPARQL to infer the knowledge base. A technology-independent API is also provided by SEANET to implement autonomic network management tasks for customers without knowledge of semantic web or telecommunication network. Details are given in Chapter 4.

A new SDN network data modelling approach "ReasoNet" [169], adopts concepts of ToCo to model their knowledge base based on Ryu controller (a SDN controller). It could support network knowledge inference and integrity/consistency validation. Two popular control applications, a learning switch application and a QoS-oriented declarative policy engine, are presented to demonstrate the scalability which is comparable with current SDN network operation systems.

In lost silence, a shipwreck early detection service [39], a methodology is demonstrated to detect shipwreck incidents immediately (with the delay in the order of milliseconds), by processing semantically annotated streams of data in wireless telecommunication systems. The evaluation results show that with a properly chosen window size, such incidents can be detected effectively and efficiently. Full details are presented in Chapter 6.

3.4 Conclusions

In this Chapter, we have developed the ToCo ontology. It provides an abstract conceptual framework for hybrid telecommunication networks – the devices, interfaces, links inside the telecommunication system, and the measurement of the link properties (or in other term, QoS, without technology specificity. The information of users and services are represented. The main concepts of a SDN are also described by ToCo.

While modelling the knowledge in networks, an ontology design pattern, the `DIL` pattern, has been observed and summarised. It provides a simple and efficient insight into the structure of ontologies for all kinds of linked devices, making the ontologies modelling process efficient, by avoiding some repetitive work.

Eight physically separated modules are arranged in ToCo, focusing on different aspects, namely, `Device`, `Interface`, `Link`, `User`, `Service`, `Data`, the key modules of ToCo are `Device`, `Interface`, `Link`. The demonstrations conducted on four networks with different technologies have shown that ToCo is able to described these networks. Concepts from existing ontologies are reused, e.g., `foaf` for user presentation, `wgs84` for location, and `UO` to describe units of measurements.

ToCo is currently used in a number of projects, SEANET, ReasoNet, Lost Silence, SARA. It is revised mainly based on the feedback from the use cases.

Chapter 4

The Proposed Knowledge Based System for SDN Management – SEANET

4.1 Introduction

In this chapter, we present our solution for the knowledge based system for SDN management – SEANET, an autonomic, lightweight, technology-independent agent of semantic network management system for SDNs. SEANET abstracts details of managing SDNs into a formally defined ontological knowledge base augmented by inference rules. The machine-understandable knowledge base allows the SEANET system to collect, process, infer, and operate on telecommunication networks and the information associated with them, in an autonomic fashion. With the developed open API, SEANET enables researchers to develop their semantic-empowered applications on their production networks.

SEANET consists of three components: a knowledge base generator, a SPARQL engine, and a Linked-Data API. The knowledge base generator can collect resources from the network and by mapping them with the ToCo ontology, a knowledge base is generated for the network. It is technology independent, and could work on networks with hybrid technologies and various topologies. The function of the SPARQL engine

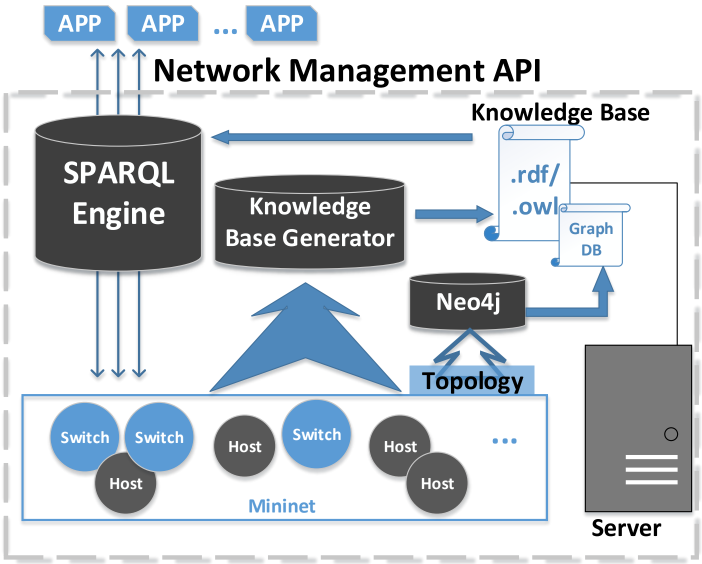


FIGURE 4.1: Architecture of the SEANET, with the proposed components: knowledge base generator, SPARQL engine and a network management API over the Mininet framework.

is to run the query on the generated knowledge base, which is also network technology-independent. Its input is a query string, and the output is the formatted results. To make the functions of SEANET available for users without knowledge of either telecommunications or Semantic Web knowledge engineering, an API is provided, offering a range of functions from knowledge base generation, query execution, to network management tasks executing. The given evaluations allow us to prove the practicability of SEANET as a semantic, high-level autonomic network managing.

4.2 SEANET Architecture

The overall architecture of the proposed system is shown in Fig. 4.1. The core SEANET functionality is enabled by extensions of the SDN emulator – Mininet[1] [155], [170], providing integration with the Neo4j[2], a graph database tool. The knowledge base generator and SPARQL engine also take advantage of command line tools provided by Open vSwitch (OVS)[3].

[1] http://mininet.org/
[2] See https://neo4j.com for detail.
[3] A multilayer virtual switch. See http://openvswitch.org for details.

4.2.1 Knowledge Base Generator

The knowledge base generator is designed to bring knowledge base harmonisation into reality, by retrieving unstructured data from nodes in the network and translating and formatting them into the semantic knowledge base. The main tools we adopts in the generator to retrieve information are *OVS* (for flows editing), *Mininet API* (for topology information retrieving), and *iperf* (for bandwidth measurement), etc. The algorithm is outlined as follows.

- First, it collects information (concepts and relations) from network elements, e.g., switches, hosts, ports, and flows in each switch.
- With the assistance of the ontology ToCo, this information is translated into RDF triples, which are semantic-enriched and machine-processable.
- The RDF graphs generated are stored as a triple-store, written into an .rdf/.owl/.ttl file on the server, and evolves automatically in real time.

The knowledge base generator adopts a recursive algorithm to read the network records through Mininet API, extracts the semantic information, and stores it to the SEANET knowledge base, to get an updated knowledge base in real time. When a network has been initiated, the modeller can be started by a simple shell command "$ py getSemanticKB.py". Information will be transformed into ontology formalised triples. These triples are stored as an ontology file in the server. This procedure is running periodically (at the evaluation process, we run it every 5 seconds, however, this value is customer configurable), maintaining an updated knowledge base for the network.

4.2.2 SPARQL Engine

The knowledge base generated by the knowledge base generator is a structured .owl/.rdf/.ttl file, which is computer processable and semantic query-ready. Semantic queries on the network knowledge base are designed to answer high level questions such as, "Which switch is host1 connected to?", "Find me the hosts in the network which are set to drop their flows." or "Find me all the hosts connected to switch 1 and switch 3, if they are not host 3 or host 5," as shown in Algorithm 1. Algorithm 1 provides the SPARQL query to get all the hosts of switches "s1" and "s3", except the hosts "h3" and "h5", in which host "h3" is a host connected to switch "s1" and host "h5" to switch "s3". SPARQL is a RDF query language. It is the tool to manipulate data stored in knowledge base formatted by ontology. As RDF describe data with "subject – predicate – object" triples, SPQRAL query use that syntax to retrieve data as well. The variable are started with "?", for example "`?p2`", "`?macAddr1`", "`?p4`", "`?macAddr2`" in Algorithm 1 are all variables, and concepts and relations in the knowledge base are expressed in the form of "`prefix:conceptName`", for example, "`net:Host`" denotes the concept "`http://purl.org/toco/Host`" with the namespace "`http://purl.org/toco/`" replaced by shortcut prefix "`net`". The "where" clause is the matching criteria, while the "select" clause list all the variables that we are seeking values for. For example, the triple in the first sentence of "where" clause: "`?p1 net:isIn net:s1`" means "find all the ports in s1, as the value of variable ?1". A complete triple is ended with period, while more than one triples share the same subject will end with semi-comma. For example, the second triples in "where" clause: "`?l1 net:linkTo ?p1; net:linkTo ?p2. filter (?p1 != ?p2).`" means "?l1 is a link, and it links to two different ports ?p1 and ?p2", in which "`filter`" is a control operator provided by SPARQL. There are many powerful operations provided by SPARQL, e.g., `JOIN, FILTER, BIND, etc.`

With semantic queries, network administration can obtain the knowledge of the network at a highly abstracted level, and thus can issue tasks from an abstract level and leave the detailed technology-specific operations to the autonomic management system.

4.2.3 Network Management API

With the knowledge base generator and SPARQL engine, it is possible to achieve a self-managed network. However, the requirement of expertise in Semantic Web and vendor

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX net: <http://purl.org/toco/>
PREFIX pos: <http://www.w3.org/2003/01/geo/wgs84_pos/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?p2 ?macAddr1 ?p4 ?macAddr2
WHERE {
  ?p1 net:isIn net:s1. ?l1 net:linkTo ?p1; net:linkTo ?p2. filter (?p1 != ?p2).
  ?p2 net:isIn ?h1. ?h1 rdf:type net:Host. filter(?h1 != net:h3).
  ?p2 net:hasMAC ?macAddr1.
  ?p3 net:isIn net:s3. ?l2 net:linkTo ?p3; net:linkTo ?p4. ?p4 net:isIn ?h2.
  ?h2 rdf:type net:Host. filter(?h2 != net:h5)
  ?p4 net:hasMAC ?macAddr2. filter (?p3 != ?p4).
}
```

ALGORITHM 1: SPARQL query to get all the hosts of switches "s1" and "s3" except the hosts "h3" and "h5", in which host "h3" is a host connected to switch "s1" and host "h5" to switch "s3".

specific network configuration has proved to be a barrier to existing systems [12], [14], [21], [58], [62], [133], [137], [137]–[143], [143]–[146].

Our another contribution is an network management API, providing basic network management services as methods. With this API, all the technology-specific details are sealed, only the technology-independent methods are exposed. Some example methods exposed by the network management API are shown in Table 4.1. Each method provided by the API is implemented as a SPARQL query with customer configurable parameters, following the Linked-Data API approach [99], [101]. For example, API method *"addFlow"* can add a flow entry to the flow table in a specific switch, and *"dump_all_flows"* can list all the flow entries of the switch. The method *"connectAll"* provides a relatively more complex function to add connections between all the hosts in a network, forming a mesh network. The method *"buildFirewall"* can build a customised firewall to block specific packages between specific nodes. Execution results of these methods are presented and discussed in Section 4.4.3.

4.2.4 Software Environment

In the development of SEANET, the following software tools are used:

TABLE 4.1: Examples of the Methods in Network Management API.

Method Name	Parameters	Function
addFlow	***dst***: the destination MAC address of the flow entry, ***in port***: the input port id of the flow entry, ***action type***: action type of the flow entry, could be output, drop, etc., ***to port***: the output port id of the flow entry.	Add a flow entry to the switch's flow table.
deleteFlow	***list of input hosts***: delete the flow from these hosts, ***list of output hosts***: delete the flow forwarding packets to these hosts	delete a flow from the switch's flow table
addARPFlow	None	Add an ARP flood flow entry to the switch's flow table, if needed.
dump_all_flows	None	Automatically list all the flow entries of the switch.
connectAll	***list of hosts***	Automatically add flow entries for all the hosts
buildFirewall	***list of switches***: the switches between which the firewall is built; ***list of hosts:*** the hosts that should be still connected after the firewall is built.	Automatically build a firewall between the given switches, keep the hosts connected in the parameter if given.
findPath	***list of hosts***	Automatically find the shortest path between the hosts.

VirtualBox: version 5.1.4.[4] To provide a virtual Ubuntu environment for the network and controller to run;

Mininet: version 1.9r2.[5] To simulate the SDN topology;

rdflib: version 4.2.1.[6] A Python library to work with RDF;

Neo4j: version 1.0.2.[7] A graph database storage and process engine.

OVS: targeted at highly dynamic, multi-server virtualization environment, OVS[8] is a multi-layer virtual switch [156]. By trying to reuse the code as much as possible, OVS provides a plethora of powerful API s for autonomic network control, and thus provide high abstract of the dynamic networks. It is worth

[4] http://virtualbox.org
[5] http://mininet.org
[6] http://rdflib.readthedocs.io/en/stable/#
[7] https://neo4j.com
[8] http://www.openvswitch.org

mentioning that, for the Linux systems later than 3.3, OVS is provided as a part of the kernel, which make the application developing with OVS much more convenient.

4.3 Rules

The integration of an ontology and policy-based network management has been intensively studied recently [14], [15], [61], [64], [134], [141], [157], and various approaches have been proposed. Policy-based network management systems view the network as RDF triples and the rules of plausible inference that connect them. A policy consists of one or more rules, and rules are logical conditions that help to identify potential knowledge that could be shared from one domain to the other. Two rules adopted in this work are given in the following as examples.

Learning switch rule: As the basic function of a network, the learning switch rule will explore the network topology, find paths for every host, and configure each switch automatically to achieve connectivity. The rule in first order logic is shown below. It says that for all the hosts h1 and h2, that connects to the Switch s, get their MAC address and add flows between them, making them be able to communicate with each other.

$$\forall h1\ \forall h2 (Switch(s) \wedge hasHost(s, h1) \wedge hasHost(s, h2) \wedge \\ isNot(h1, h2) \wedge hasMAC(h1, m1)\ \wedge hasMAC(h2, m2) \rightarrow addFlow(m1, m2))$$

Automatic flow update rule: To accomplish an autonomic network management system, the system needs to be self-aware. The automatic flow update rule is able to detect any port failures, update the path in the network, and reconfigure the corresponding flow entries in real time. The rule in first order logic is shown as below. It says that for all the flows in a switch s, if the port this flow is trying to forward to is down, then this flow is inconsistent with the real status of the network.

$$\forall f\ (Switch(s) \wedge PathFlow(f) \wedge hasFlow(s, f) \wedge$$
$$in\_port(f, in\_port) \wedge hasAction(f, a) \wedge toPort(a, to\_port) \wedge Port(in\_port) \wedge$$
$$isUp(in\_port, \text{“}false\text{''}) \rightarrow inconsistantFlow(f))$$

The SPARQL query for automatic flow update rule is shown in Algorithm 2. It has been evaluated on the following sections. Any non-empty query results returned by the algorithm denotes that there is a inconsistency flow exists.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX net: <http://purl.org/toco/>
PREFIX pos: <http://www.w3.org/2003/01/geo/wgs84_pos/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?in_port ?to_port
WHERE {
  ?s a net:Switch; net:hasFlow ?f.
  ?f a net:PathFlow; net:in_port ?in_p; net:hasAction ?a. ?a net:toPort ?to_p.
  ?p1 a net:Interface; net:hasInterfaceName ?in_port; net:isUP ?isUp1.
  ?p2 a net:Interface; net:hasInterfaceName ?to_port; net:isUP ?isUp2.
  filter (?isUp1 = "false"^^xsd:boolean || ?isUp2 = "false"^^xsd:boolean)}
```

ALGORITHM 2: SPARQL query for automatic flow update rule. A non-empty result returned by the query denotes the rule has been violated.

4.4 SEANET Functionalities Demonstrations

We evaluate SEANET by carring out different networks generated by Mininet, with the topologies of linear, single, and tree, and the network scales vary from small (3 hosts) to large scale (≥1000 hosts).

4.4.1 Hardware Environment

The mechanism and demonstrations are carried out on a MacBook Air, OS X 10.9.5, Intel Core i5, 1.5 GHz, 4GB RAM, Ubuntu 14.04 on VMware workstation. The VMware system has 1GB RAM.

4.4.2 Evaluation of the knowledge base generator

We have run the knowledge base generator on different networks with the topologies of linear, single, and tree. The largest network is the one with tree topology, depth = 3, fanout = 5, which contains 1146 nodes (1021 switches and 125 hosts). The smallest network is a single network with $k = 5$ (k is the number of hosts connected to each switch), which has 6 nodes (1 switch and 5 hosts) in total. The topology of a network with tree topology, depth = 3, fanout = 3 is shown in Fig. 4.2. Fig. 4.3 shows the screenshot of the knowledge base representation for a simple network with hybrid technologies that was generated by Mininet-WiFi[9]. The results of running the generator have been analysed using the metric of response time and the number of RDF triples generated, as shown in Fig. 4.4 and Fig. 4.5. In Fig. 4.4 the scale of the networks are controlled through k (the number of hosts connected to each switch) in single and linear topology, or *depth* and *fanout* in tree topology. The scale of the networks are shown in Fig. 4.5 by the diameter of the circles, varies from a small network with 6 nodes to a large one with more than 1000 nodes. The longest response time, 2.591602s, is taken by the second largest network (linear with $k = 100$), which has 200 nodes and 199 links. As illustrated in both Fig. 4.4 and Fig. 4.5, the response time of knowledge base modelling increases linearly both with the increase of the number of network's nodes and the number of RDF triples. However, one abnormality (the orange dot) is shown at the up-right corner in Fig. 4.5. It denotes a network whose modelling process is much slower than those with the same scale. It is mainly due that although it is small in scale, but the RDF triples required to describe it is much larger than the others. Thus, it seems that the number of RDF triples required, rather than the scale of the network, that has a decisive effect on the speed of semantic knowledge base construction.

[9]https://github.com/intrig-unicamp/mininet-wifi

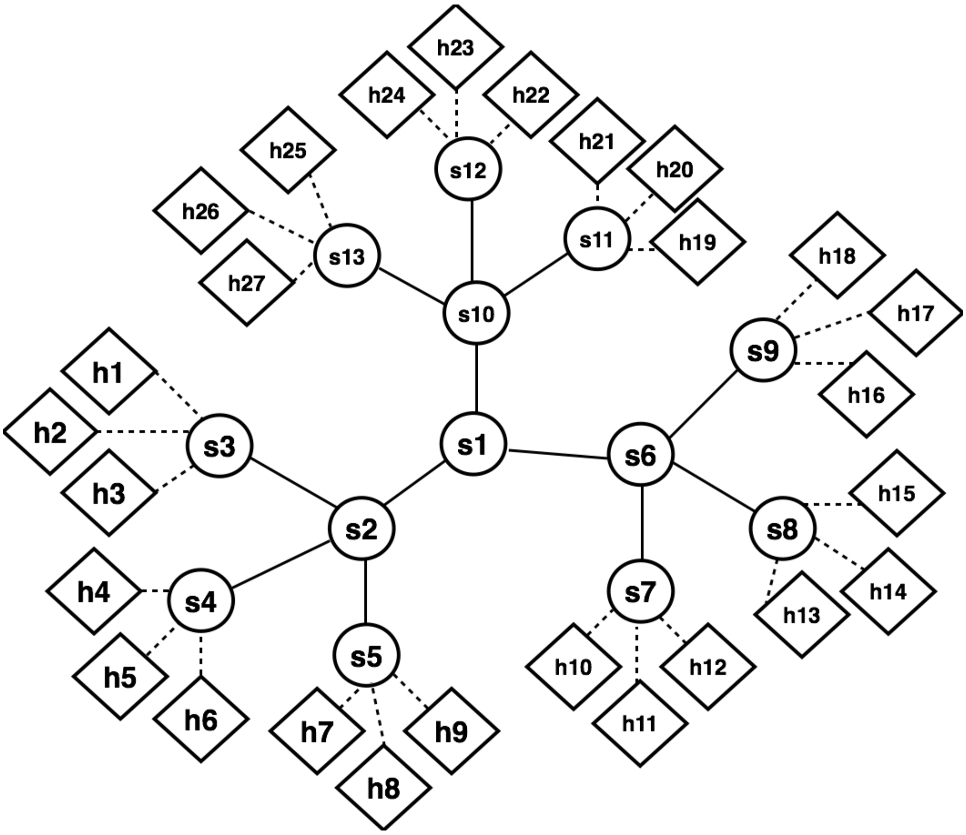


FIGURE 4.2: The topology of a network with tree topology, depth = 3, fanout = 3.

4.4.3 Evaluation of the SPARQL engine and Network Management API

As the network management API is driven by the SPARQL engine, the corresponding SPARQL queries are shown in Algorithm 1 and Algorithm 2, and the execution results are shown in the demo of the network management API.

To evaluate the performance of the API, we executed the following methods to evaluate how the basic network management tasks (such as adding a flow, listing all the flows of a switch) and complex tasks (such as connecting hosts to network, and building a firewall) can be accomplished with a single line command.

- *"addFlow"*: Add a flow entry to the flow table in a specific switch.
- *"dump_all_flows"*: List all the flows in a switch.

```
@prefix ns1: <http://home.eps.hw.ac.uk/~qz1/> .
@prefix ns2: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ns1:Link0 a ns1:Link ;
    ns1:hasStatus "OK"^^xsd:string ;
    ns1:linkTo ns1:h1-eth0,
        ns1:s0-eth1 .

ns1:Link1 a ns1:Link ;
    ns1:hasStatus "OK"^^xsd:string ;
    ns1:linkTo ns1:ap1-eth2,
        ns1:s0-eth2 .

ns1:Link2 a ns1:Link ;
    ns1:hasStatus "OK"^^xsd:string ;
    ns1:linkTo ns1:ap1-eth3,
        ns1:ap2-eth2 .

ns1:Link3 a ns1:Link ;
    ns1:hasStatus "OK"^^xsd:string ;
    ns1:linkTo ns1:s0-eth3,
        ns1:sate10-eth2 .
```

FIGURE 4.3: Screenshot of a knowledge base generated. It is actually a .ttl file generated by the knowledge base generator. The format "turtle" is adopted when writing the knowledge graph into the file. The other options for the format are "rdf", "trix", "xml", "nt", "pretty-xml", etc. The only difference of these formats is the syntax displayed in the file.

- *"connectAll"*: Connect all the hosts in a network automatically, the SPARQL query is shown in Algorithm 1.

- *"buildFirewall"*: Build a customised firewall between two or more switches. Block packets between certain hosts, while letting the other hosts under these switches pass.

As the first experiment, we set up a network with a switch "s1", and add a flow to switch "s1" with the API method *"addFlow"* and illustrated the results with the method *"dump_all_flows"*. Execution results are shown in Fig. 4.6, in the form of a screenshot.

In the second experiment, we take methods "connectAll" and "buildFirewall" from the Linked Data API as examples to demonstrate the time efficiency, operation efficiency,

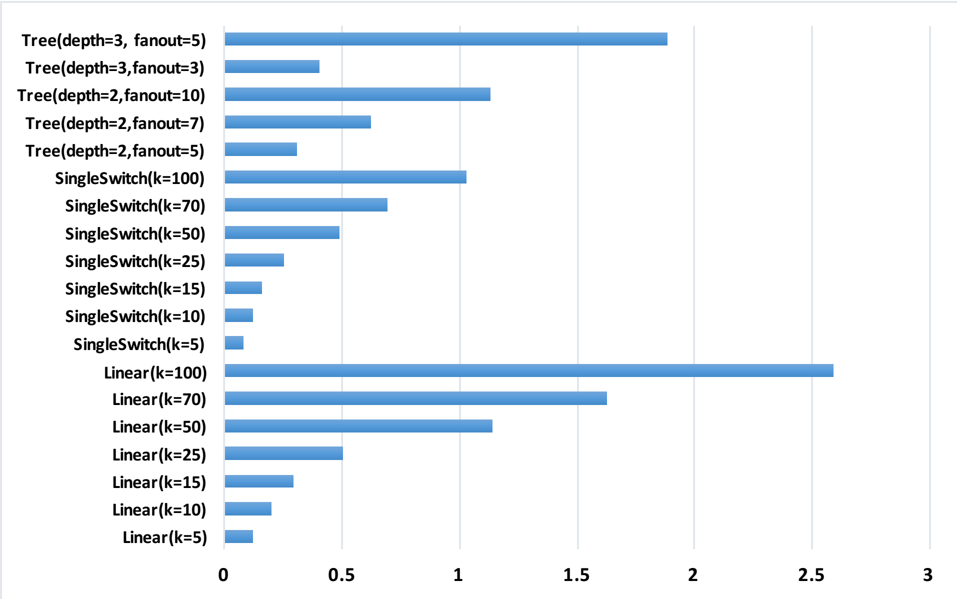


FIGURE 4.4: Response time of generating knowledge base models for networks with different topologies and scales. It is defined by the time from the start of knowledge base generator, until the knowledge graph generated is written in the file on the server. The scale of the networks are controlled through k (the number of hosts connected to each switch) in single and linear topology and denotes the number of hosts connected to each switch, or depth and fanout in tree topology. As shown in the figure, the time grows linearly with the increase of number of nodes.

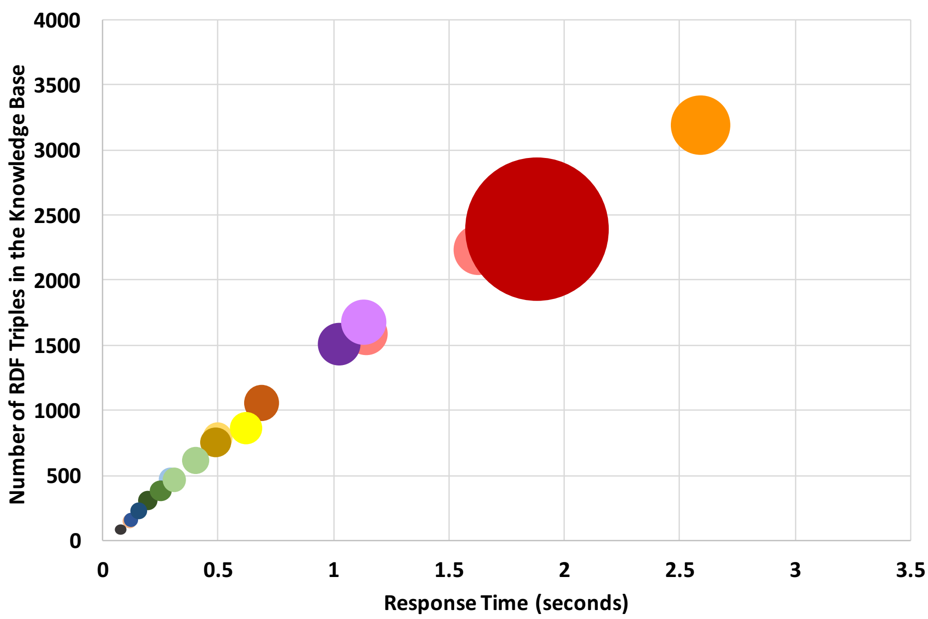


FIGURE 4.5: Response time of knowledge base modelling for different networks and the number of RDF triples generated. A dot represent the experiment of a network. The scale of the network is denoted by the diameter of each dot. As shown in the figure, the time grows linearly with the number of triples in the knowledge base.

```
>>> from api.my_ovs_switch_API import switchAPI
>>> a = switchAPI('s1')
>>> r = a.dump_all_flows()
NXST_FLOW reply (xid=0x4):

>>> r = a.add_flow(dl_dst='00:00:00:00:00:01',actions='drop',priority=100)

>>> r = a.dump_all_flows()
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=15.049s, table=0, n_packets=0, n_bytes=0, idle_age=15,
 priority=100,dl_dst=00:00:00:00:00:01 actions=drop
>>>
```

FIGURE 4.6: Screenshot of the execution result of API method "addFlow" and "dump_all_flows", which add a flow entry to the flow table in switch "s1", dropping all the packages with the destination MAC address of "00:00:00:00:00:01". The execution results of the API method "dump_all_flows" show that the flow has been added successfully.

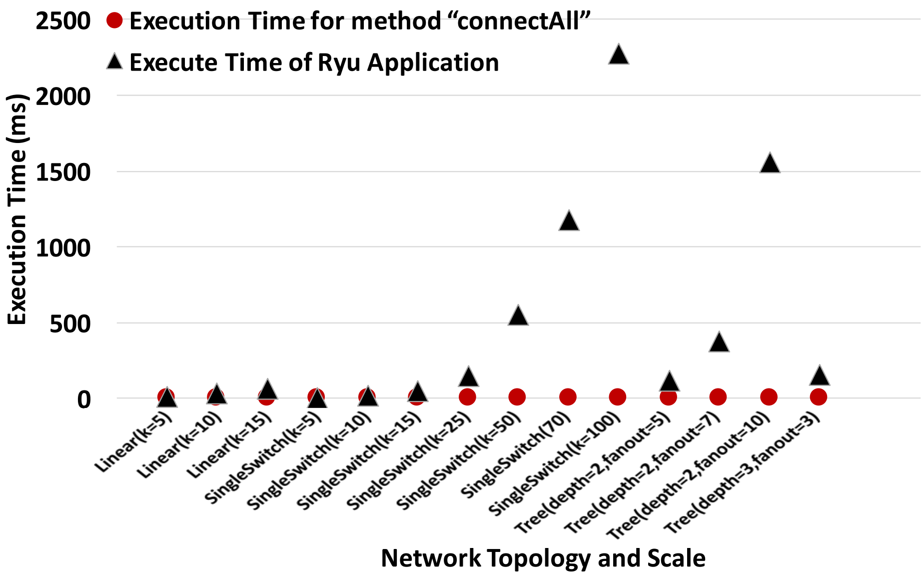


FIGURE 4.7: Execution time of network management API method "connectAll" and Ryu App with the same function, tested on networks with different topologies and scales. A black triangle represents the execution time of Ryu App, and the red dot represents the time of our API method "connectAll" on the same network. As shown in the figure, the comparison is quite obvious. Although the times are similar when the network is small (< 15 nodes), it soars violently as the scale of the networks increase. This phenomenon can be recognised in networks with all kinds of topology. In a single switch network with 100 nodes, our method is 2300 more efficient than Ryu application.

and code length, comparing with one of the most popular SDN controllers – Ryu[10]. These methods are run on networks with various topologies and scales, from simple network with 5 hosts to large network with 1146 nodes, as shown in Fig. 4.7, Fig. 4.8.

[10] https://osrg.github.io/ryu/

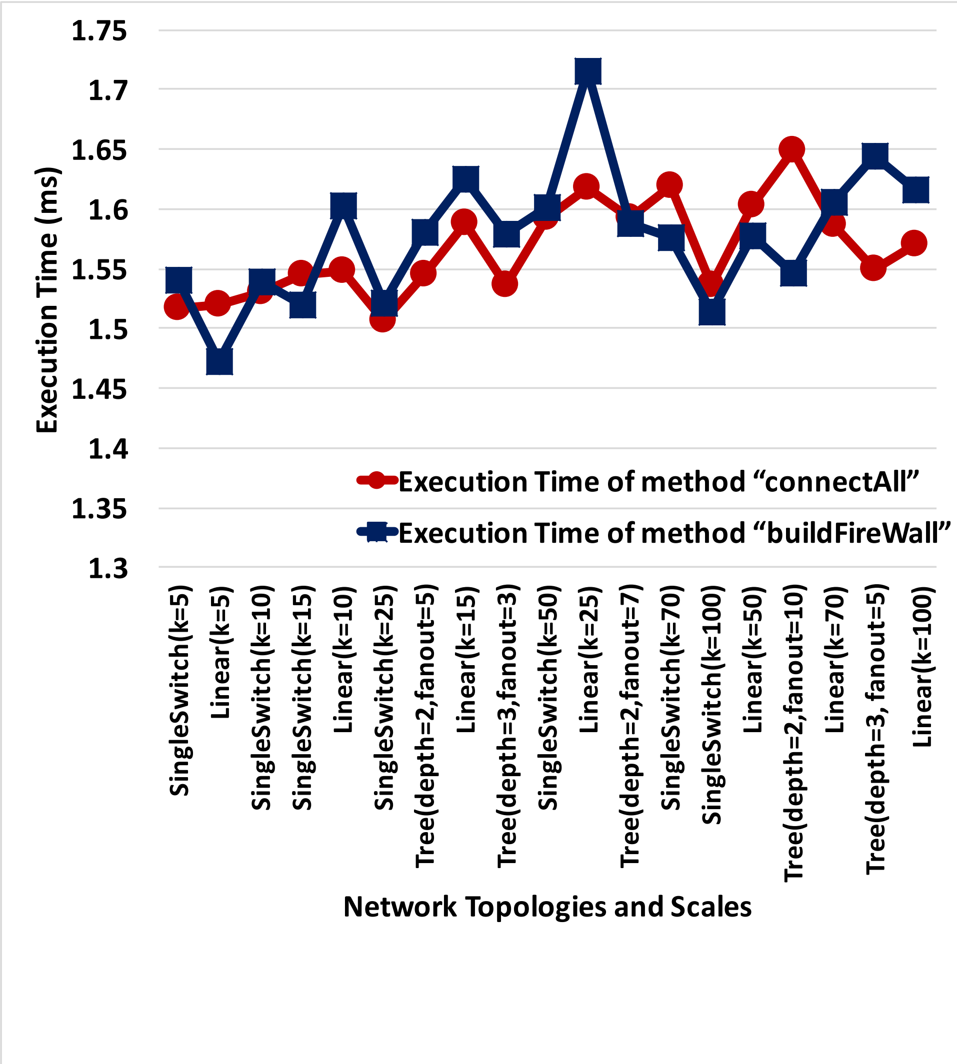


FIGURE 4.8: Execution time of the methods "connectAll" and "buildFireWall" from the network management API, tested on networks with different topologies and scales.

The method "connectAll" is compared with an existing application from Ryu. With "connectAll", all the hosts inside the network can be connected with each other, regardless of the network topology and complexity. The application provided by Ryu, "simple switch.py", accomplishes the same function. The comparison between our network management API and the Ryu application in terms of execution time on the same networks is shown in Fig. 4.7. Although the execution time of Ryu application is equivalent to our method in small networks (with less than 15 nodes), Ryu experiences immense increases as the network scale grows. The method "connectAll", however, has a stable performance. This is probably because of the code framework of Ryu is over

Table 4.2: Comparison result of network management API "connectAll" and a ryu application with the same function.

	Ryu App "simple switch.py"	**SEANET API Method "connectAll"**
Number of Lines in the source code	120 lines	5 lines for the query code
Code reusability	Poor. It is not an API, which means it is not reusable.	High. Plug-and-go and reusable.
Code Readability	Poor. Developers have to be quite familiar with Ryu architecture.	Easy. API method call SPARQL query knowledge

complicated (adopting too many Python decorators[11], as discussed later in Table 4.2). Another reason may lays in that the execution on the Semantic Web knowledge base mainly depends on the size of the knowledge base, and grows linearly with the size of the knowledge base, as illustrated by the above evaluation results of knowledge base generator. Thus it is more efficient than Ryu in large scale networks. In large networks where the number of nodes reach to more than 100, the method "connectAll" is 2300 times more efficient than the corresponding application in Ryu. The code complexity is also compared between "connectAll" and Ryu application, as shown in Table 4.2, in terms of code length, code reusability, and code readability. It illustrates that our method saved up to 99.9% execution time compared to the corresponding Ryu application, with only 1/24 the length of code, and better readability and reusability. This is because the network management API in SEANET depends on the SPARQL engine to reason over the knowledge base, which is more efficient and stable, comparing with Ryu, which based on traditional databases.

We have also compared the complexity of the "buildFirewall" method with a Ryu application completing the same task, building a customised firewall between any nodes, the result is listed in Table 4.3. Ryu's RESTful API is adopted, but only low level methods are provided by the API, which make the commands in Ryu tedious, error prone, and requires specific knowledge of the network and programming to operate. The "buildFirewall" method provided by the SEANET API, however, is high level,

[11]https://wiki.python.org/moin/PythonDecorators

Table 4.3: The commends for building the same firewall with Ryu and SEANET.

Commends required by Ryu	**Commend required by SEANET**
> xterm c0 # ovs-vsctl set Bridge s1 protocols=OpenFlow13 # ryu-manager ryu.app.rest_firewall # curl -X PUT `http://localhost:8080/firewall/module/enable/0000000000000001` # curl -X POST -d '{"nw_src":"10.0.0.2/32","nw_dst":"10.0.0.3/32"}' `http://localhost:8080/firewall/rules/0000000000000001` # curl -X POST -d '{"nw_src":"10.0.0.3/32","nw_dst":"10.0.0.2/32"}' `http://localhost:8080/firewall/rules/0000000000000001` # curl -X POST -d '{"nw_src":"10.0.0.2/32","nw_dst":"10.0.0.3/32", "nw_proto": "ICMP","actions":"DENY","priority":"10"}' `http://localhost:8080/firewall/rules/0000000000000001` # curl -X POST -d '{"nw_src":"10.0.0.3/32","nw_dst":"10.0.0.2/32", "nw_proto": "ICMP", "actions": "DENY", "priority":"10"}' `http://localhost:8080/firewall/rules/0000000000000001`	# firewall(s1, s2, host1=h1, host2=h2)

simply one line and self-explaining. When executing the "buildFirewall" method, the following commands will be executed automatically.

1. Compose a SPARQL query string customised by the given parameters automatically, to find the MAC addresses of the hosts that the firewall will be built between.
2. Execute the SPARQL query automatically.
3. Based on the query results, add flows indicating drop packets to/from these MAC addresses in the query results.

These operations will be executed autonomically. Thus, the user does not need to be an expert in network or programming to execute it. The contrasting of technology intensive and technology independent operation between SEANET and Ryu is particularly clear.

4.5 Evaluation by Use Case

This section describes how we evaluate the SEANET knowledge base in a common network management scenario of automatic flow updating. An autonomic network must be self-aware. Thus, it should be able to learn what is happening inside, detect changes, decide what to do, and fix the problem itself. We choose the scenario of inconsistent flow detection. In SDN, flows are adopted to route packets to/from specific port. If a port accidentally fails, the flows related (the flows with instructions to send packet from/to this port) should be revised (stop sending packets to this failed port). In an autonomic network, the controller should be able to discover this autonomically. In the following experiments, we demonstrate how the network can sense and react on random node failures, adopting the SEANET knowledge base. The automatic flow update rule is adopted. The evaluation is designed to investigate if it is possible to detect all the accidental port failures and all the related inconsistent flows. The evaluation assesses the performance in terms of reaction time.

4.5.1 Scenario

When networks are initialised, paths will be automatically found for all the nodes in the network, and then flows will be automatically created for each path. At randomly chosen times, randomly chosen ports disabled or recovered. In the scenario, the metrics that are used to measure the performance are the number of flows detected that are inconsistent with the current network status (that is, the ports forwarded to or from by the flows that are down), and the response times, i.e., how quickly the failure is detected.

4.5.2 Simulation and Execution

We run a series of 3 tests, with varying network port failure states. The network is emulated by Mininet-WiFi, which has linear topology with three switches and three hosts. In each test, the automatic flow update rule is run every 5 seconds for

200 seconds, watching the status of the network. Tests are executed in a simulated environment. The goal is to measure the number of inconsistent flows detected and the response time from when the ports failed. We use an emulation-based evaluation with Mininet-WiFi, Ryu, and the semantic tool – rdflib. Simulations are carried out on a Dell PowerEdge R630 Rack Server with 1.5TB memory, driven by two Intel(R) Xeon(R) E5-2600 v4 processors, operating at 3.60GHz.

The SPARQL query string is shown in Algorithm 1. The rule is running every 5 seconds, watching the status of the network. We bring randomly chosen ports down and up at random times, and measure how many affected flows are detected and the reaction times. The results for two different tests are shown in Fig. 4.9 and Fig. 4.10. In Fig. 4.9, three ports are taken down sequentially at time slots: 22.51 milliseconds – 65.23 milliseconds, 137.63 milliseconds – 172.37 milliseconds, and 189.17 milliseconds – 226.05 milliseconds. In Fig. 4.10, three ports are taken down at time 5.61 milliseconds, 19.98 milliseconds, and 40.01 milliseconds, and then brought up at time 88.67 milliseconds, 115.32 milliseconds, and 132.28 milliseconds, respectively. The number of inconsistent flows detected and the number of ports down are shown by the lines with different colours (orange for ports, blue for flows). As shown in Fig. 4.9 and Fig. 4.10, the system captures every port failure successfully and efficiently, not only in the single port failure situation, but also in the situation that multiple ports failed simultaneously. The figures also show that the system reacts swiftly when ports recover from failure, in both situations of single port failure recovery and multiple port failure recovery. According to the definition of OpenFlow, for one hop in a path "port001" – "port002", there will be two flows defined with the opposite directions. Thus, if one of these two ports fail, there will be two flows that are affected. As shown in Fig. 4.9 and Fig. 4.10, when there are one, two, and three port failures, the system detects two, four, and six flows that are affected respectively, which demonstrates the feasibility of the rule execution.

The response times of the rule executions are further investigated, as shown in Fig. 4.11 and Fig. 4.12. The response time of one query execution is shown as the delay of the query outcomes after the port is down in Fig. 4.11, (the details of the query is given

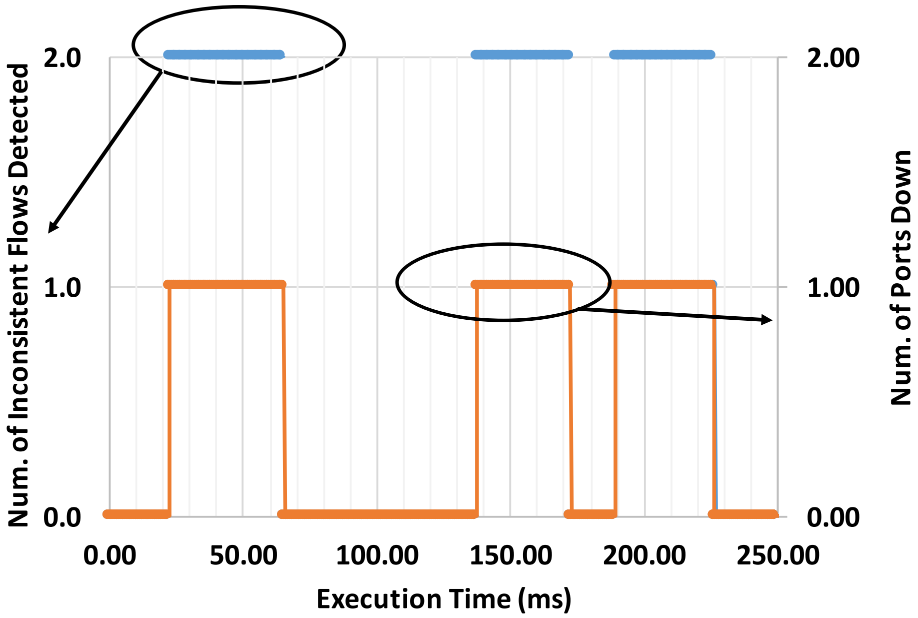


FIGURE 4.9: Evaluation results of a test, in which three ports are taken down and up at 22.51ms-65.32ms, 137.63ms-172.37ms, and 189.17ms-226.05ms, respectively. The orange curve denotes the number of ports that are down, the blue line denotes the number of inconsistent flows detected.

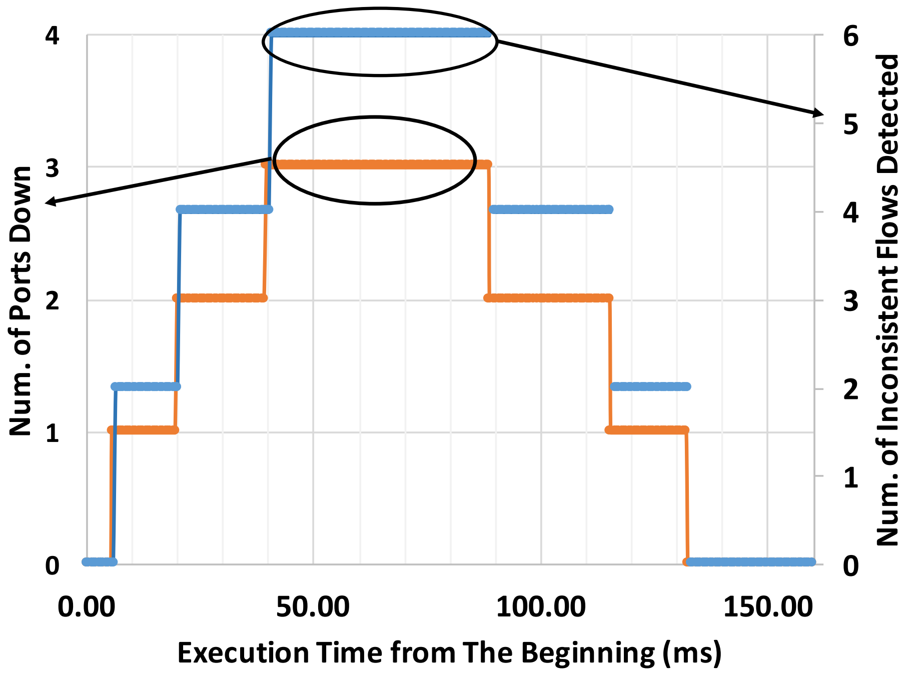


FIGURE 4.10: Evaluation results of a test, in which three ports are taken down at time 5:61 ms, 19:98 ms, and 40:01 ms, and then brought back up respectively at time 88:67 ms, 115:32 ms, and 132:28 ms. The orange curve denotes the number of ports that are down, the blue curve denotes the number of inconsistent flows detected.

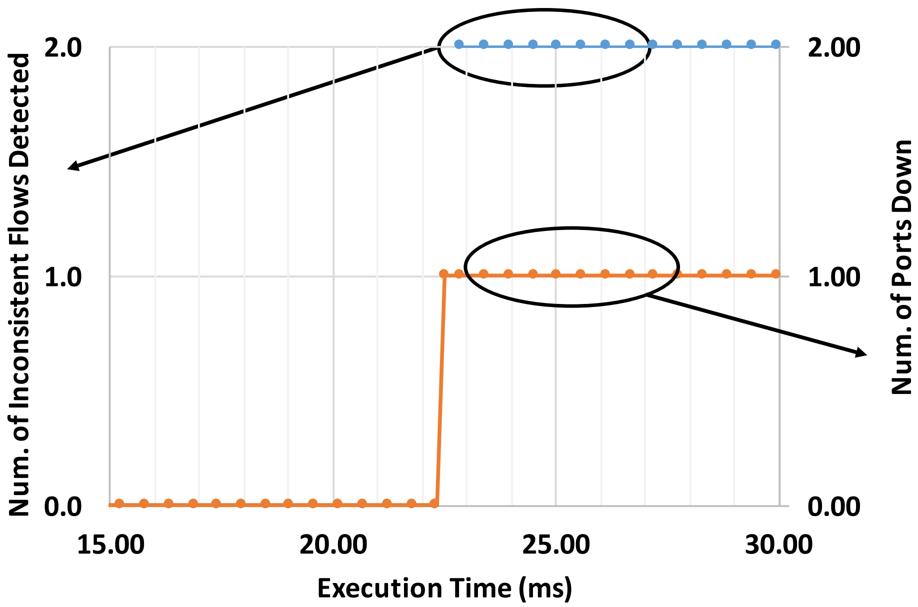


FIGURE 4.11: Response time shown in the evaluation result of the test 1. The rule execution response (the blue curve) has a small delay (less than 0.5 ms) comparing with the ports' status change in the network (the orange curve).

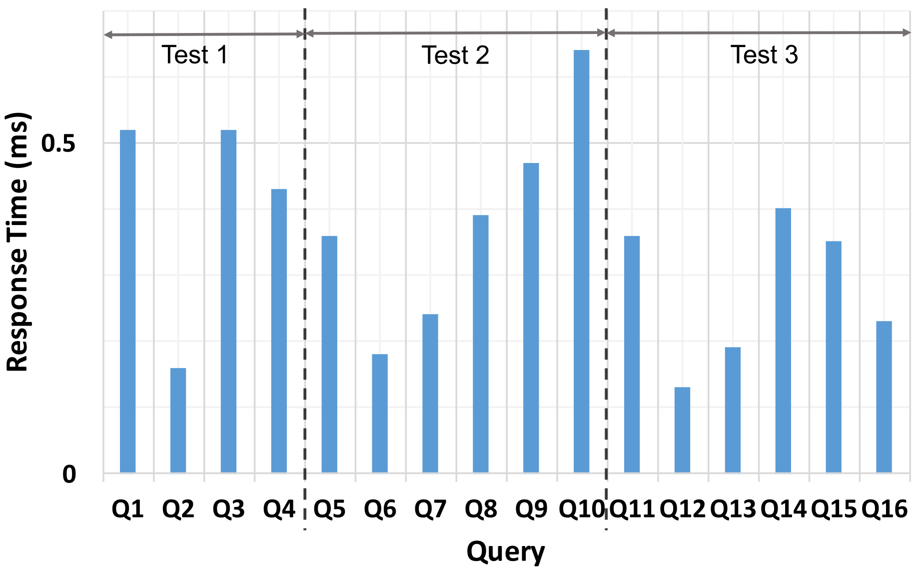


FIGURE 4.12: Response times of all the queries in the three experiments, as shown in the figure, Q1 – Q4, Q5 – Q10, and Q11 – Q16, respectively. The response time is generally smaller than 0.5 milliseconds. The average response time round up at about 0.3 milliseconds, which is acceptable for autonomic network management.

in Appendix C). We execute the experiment three times in the same network and summarize all the response times in Fig. 4.12. The results of Q1 – Q4 are from the first test, Q5 – Q10 come from the second test, and Q11 – Q16 from the third test. As shown in Fig. 4.11, the response time of the query execution is less than 0.5 milliseconds. The response times for every query in Fig. 4.12 further prove that. Most of the reaction time are below 0.5 milliseconds, and the slowest execution takes only 0.6 milliseconds.

4.6 Conclusions

In this chapter, we have proposed SEANET, an ontology-based solution for autonomic management system of SDN. The heart of this system is a knowledge base for SDNs that makes autonomic network knowledge inference possible. SEANET consists of three parts, namely the knowledge base generator, the SPARQL engine, and the network management API. The knowledge base generator can generate a semantic-enriched knowledge base of a network automatically. The SPARQL engine can apply queries on the knowledge base. The network management API works as an interface that can translate highly abstracted, technology independent method calls into technology-specific operations.

Evaluations have been carried out on all three parts, based on different network topologies and scales. The performance has been analysed using the metrics of response time. The functionality of SEANET was also shown through a use case – forwarding policy conflict detection. In summary, many complicated tasks, e.g., connecting one or more hosts to a switch, adding one or more flow entries to a flow table, or even building a firewall, could be accomplished in one single method call and the programs could execute the task automatically, which is the ultimate goal of SDN.

Chapter 5

SARA – Semantic Access Point Resource Allocation Service

This chapter and the following chapter will shed light on the use cases of our proposed knowledge based network management system – SEANET. The previous chapter has evaluated some basic network management tasks ranging from routing, finding a path, building a firewall. This chapter and the next one will further demonstrate use cases such as macroscopic network management decision making service (e.g., resource allocation in heterogeneous access technologies) and disaster response applications (e.g., tragedy early detection).

5.1 Introduction

Current wireless networks have hybrid access technologies coexisting together [171]–[174], e.g., WiFi, Long Term Evolution (LTE), 3G/2G, Satellite, Mesh, Ad Hoc, etc. How to use these technologies effectively, e.g., choose among these technologies intelligently, allocate resource dynamically based on the customised requirement, etc., has been intensively investigated in recent times [175]–[184]. However, the technologies and devices are usually operated and supported by different companies and vendors. Usually devices produced by one company cannot use the network provided by another company. It is not realistic to apply macro-controls over heterogeneous networks with current network management system. In this chapter, we demonstrated how to integrate the resources from different access technologies, such as LTE, LiFi, WiFi, Satellite, Mesh, and allocate resources and route traffic automatically with SEANET. The work is evaluated on emulated hybrid wireless networks with different scales.

Considerable research has been devoted to the AP selection problem in a homogeneous network [175]–[184]. The approach adopted by most of the current network management systems is to simply choose the AP with the strongest signal strength. However, this signal strength strategy (SSS) does not necessarily guarantee quality of service to users, particularly in scenarios with an unbalanced load, in which the user and service demands are concentrated in a relatively small area [175]–[184]. "Virgil" is an advanced AP selection strategy was proposed by Nicholson et al. [175]. It can quickly scan and test all available APs, and select the one with the best connectivity performance. Vasudevan et al [179] considered "potential bandwidth" as the metric when choosing an access point. Although numerous strategies have been proposed for AP selection in homogeneous network, the AP selection in heterogenous networks has not been as widely investigated. A fuzzy logic approach was applied to select between hybrid LiFi and WiFi networks by Wu et al. [182]. An extended Analytic Hierarchy Process (AHP) method was adopted in [181] to select and route among the hybrid wireless networks with various technologies (including satellite, Third Generation (3G), LTE, WiMAX, and Wi-Fi). However, these methodologies only consider fixed metrics, and cannot adapt to complex customised rules.

In this Chapter, we present a Semantic Autonomic Resource Allocation Service (SARA). It demonstrates how to choose from different access technologies, such as LTE, LiFi, WiFi, Satellite, and Mesh, considering the real quality of service, and can also adapt to user defined rules by autonomic rules reasoning. It is an application of our knowledge based autonomic network management system – SEANET [167], which builds an RDF formatted knowledge base for networks with various technologies and topologies, and apply autonomic reasoning over it. The work is evaluated on emulated hybrid wireless networks of different scales.

This Chapter is organised as follows. Following a brief introduction of current access point selection strategies, Section 5.2 presents our strategy for SARA, the algorithm, the SPARQL query, and the AP selection rule are presented in detail. Experiments are carried out to evaluate our algorithm and the results are presented in Section 5.3. The target questions are raised at the beginning, followed by the introduction of the software tools used. The performance of SARA, in terms of the bandwidth improvement, as well as the overhead, in terms of the time expenses, are presented and discussed. Section 5.4 concludes this chapter.

5.2 Access Point Selection Strategy

Access point selection strategies are strategies that choose the access point that will achieve the maximum benefit. Current access point selection strategies can be summarised as follows:

Random: algorithm chooses an unencrypted AP at random;

SSS: chooses the unencrypted AP with the strongest signal strength;

Omniscient: simulates an algorithm which uses the results of AP probes to choose the AP with the best bandwidth [175].

5.2.1 SARA – Semantic Access Point Resource Allocation Service

Like other knowledge based applications, the SARA process depends entirely on the SEANET knowledge base. SEANET ensures a real-time knowledge base by enforcing a periodic update. It refreshes its knowledge base every 5 seconds by default, but this is a configurable value. Each time SEANET updates its knowledge base, the channel information of the APs is scanned and updated. The knowledge entities of each AP adopted in SARA are shown in Fig. 5.1.

Basic AP information:

name, ssid, location,
channel number, mode type, frequency,
MAC address, driver,
signal range, transmit power,
antenna gain, antenna height,
stations in range, stations associated.

For APs in range of the target station:

Bandwidth tested,
Round trip time tested.

Figure 5.1: Knowledge entities about Access Point used in SARA from the SEANET knowledge base.

The goal of SARA is to reallocate the data traffic burden to less-busy APs. We consider the following criteria in our AP selection strategy.

- ***Number of Current Users:*** the number of mobile stations associated with an AP. In our post-incident scenario, the more users associated with an AP denotes the heavier traffic load it bears;

- ***Bandwidth:*** the bandwidth between the AP and the target station, tested with *iperf* automatically by SARA;

- ***Transmit Strength:*** the signal strength the user received from the AP. It is presented as antenna gain and transmit power in the SEANET knowledge base.

SARA algorithm: SARA's algorithm for selecting a new access point is given in Algorithm 3 and works as follows:

1 Select all available access points in range.

2 Measure the bandwidth between the target station and the available access points. Update this information in the SEANET knowledge base.

3 Query the knowledge base for the following properties of these APs.

- The total number of mobile stations currently associated with the AP;
- Bandwidth between the station and these APs;
- The transmit power of the AP;
- The antenna gains of the AP.

4 Sort the APs by these three properties with following priority: total number of stations > bandwidth > transmit power > antenna gain.

5 Choose the best AP, i.e. the first in the ordered list.

This algorithm is implemented as a SPARQL query on the knowledge base SEANET build. The SPARQL query is shown in Algorithm 3. The .sparql file could be found in Appendix C.3. The line by line interpretation could be found below.

"`SELECT ?ap (COUNT(?assoSta) AS ?cnt):`" – The variables that this query will return are `?ap` and the count number of `?asso`, which denotes as another variable `?cnt`.

"`?aps :stationsInRange :sta1;`" – Find all the Access Points (`?aps`) that the Mobile Station `:sta1` is in range.

"`:associatedStations ?assoSta;`" – For all the Access Points found in the previous line, find their associated stations `?assoSta`.

"`:hasWLAN ?w.`" – For all the Access Points found, find their interfaces `?w`.

"`?w :antennaGain ?g; :hasTxPower ?tx.`" – For the interfaces `?w` just found, find their antenna gain `?g` and transmit power `?tx`.

"`?asso a :Association; :from ?aps; :to :sta1;`" – Find the link `?asso` between the Mobile Station `:sta1` and Access Points `?aps`.

"`:hasBandWidth ?bw.`" – Find their bandwidths.

"`?bw :hasValue ?bwValue.`" – Get the values of these bandwidths.

"`bind(strafter(str(?aps), ''http://purl.org/toco/'') as ?ap).`" – For simplicity, remove the namespaces of all the Access Points found, leave only the name. This is a syntax sugar.

"`GROUP BY ?aps`" – Group the query results by the Access Point `?aps`.

"`ORDER BY ?cnt DESC(?bwValue) DESC(?g) DESC(?tx)`" – Sort the query results first by the number of associated stations for each Access Points `?cnt`, from small to big; then by the bandwidth value `?bwValue`, from big to small; then by antenna gain `?g`, from big to small; at last by transmit power `?tx`, from big to small.

```
PREFIX : <http://purl.org/toco/>
SELECT ?ap (COUNT(?assoSta) AS ?cnt)
WHERE {
  ?aps :stationsInRange :sta1 ;
      :associatedStations ?assoSta; :hasWLAN ?w.
  ?w :antennaGain ?g; :hasTxPower ?tx.
  ?asso a :Association; :from ?aps; :to :sta1;
      :hasBandWidth ?bw.
  ?bw :hasValue ?bwValue.
  bind(strafter(str(?aps), "http://purl.org/toco/") as ?ap).
  }
  GROUP BY ?aps
  ORDER BY ?cnt DESC(?bwValue) DESC(?g) DESC(?tx)
```

Algorithm 3: SPARQL query to find the best AP for a mobile station "sta1" considering the number of traffic load and signal strength of the APs.

Service specific AP selection rule: In addition to the AP selection algorithm users can set up their customised rules to select specific technology for different kinds of service. An example of the rules in first order logic is shown below.

$$\forall UserEquipment(?u) \wedge hasService(?u, ?s)$$

$$\wedge isVideo(?s) \rightarrow associateTo(?u, LTE)$$

The customised rule can be reasoned automatically by SEANET. If it is enabled, the AP will be assigned according to the rule, the rest are assigned by the algorithm.

5.3 Evaluation

The application is evaluated with real traffic in emulated networks. A scenario of an incident is emulated where a crowd gathers and consequently the service requirement steeply increases. Multiple access technologies are available in our scenario, including WiFi, Satellite, LTE, and D2D Mesh network.

When evaluating our proposed implementation SARA, we tried to seek answers to the following questions:

- How often does SARA successfully reselect and reconnect an AP for users, based on given rules?

- How much better is a user's connectivity when SARA performs the AP selection?

- What is the overhead in terms of execution time? Is it acceptable?

We have tried to answer these questions based on the result of evaluation experiments. The evaluation experiment is carried out on network emulator – Mininet-WiFi [170], which is a wireless network emulator running on virtual machine. The evaluation was carried out on a MacBook Air OS X 10.9.5 with a cache of 3MB, running on Intel Core i5 at 1.5 GHz. The system has 4 GB RAM and 128 GB SSD.

The following tools were adopted to build and query the semantic knowledge base.

- ***Rdflib:*** version 4.2.1. A Python library to work with RDF and execute the SPARQL query;

The tools adopted to monitor the network throughput were:

- ***iperf:*** measure the available channel bandwidth;
- ***iwconfig:*** to configure a wireless network interface;
- ***iw:*** to show and/or monitor the wireless network devices and their configuration;
- ***ping:*** to check the connectivity of a connection and test the round-trip time.

5.3.1 Evaluation Results

We emulate a hybrid network with the access technologies of WiFi, satellite, LTE, and 3G/2G. Pressure tests were also carried out on SARA with the experiments repeated 10 times on networks of different scales, with the number of stations varying from 10 to 100.

In our experiment, APs and stations are randomly distributed in an area of $300m \times 300m$. At the beginning of the experiment, an incident occurs, and all of the mobile stations begin to gather at the scene of the incident, with all of them taking pictures, making phone calls, shooting videos and uploading many of these to the cloud. Consequently, telecommunication network traffic exhibits a extremely rapid increase, and the channel quality quickly deteriorates. At a randomly chosen time, SARA is executed on randomly chosen phones. The QoS of the channel of selected phones before and after executing SARA are observed and compared. In particular, we have calculated the bandwidth of the chosen phones, with the unit of Mbps, as a metric. The results are shown in Fig. 5.2 and Fig. 5.3.

Fig. 5.2 is the screenshot of the terminal of a mobile station "sta6", before and after SARA is executed. The *iwconfig* results show that the SARA has successfully select and re-associated "sta6" to another AP. Before SARA is executed, "sta6" is associated to a network with the ssid "satellite", after that, "sta6" is associated to another network "lte". The bandwidth of "sta6" before and after the execution is also observed and recorded in Fig. 5.3. The red dash line denotes the time when SARA was executed. It can clearly be seen in Fig. 5.3, that the bandwidth of "sta6" has greatly improved, from around 3 Mbps to roughly 9 Mbps. Thus, the channel QoS has been enhanced three times better than before SARA is executed.

5.3.2 Overhead

We collected a diverse set of data on the time overhead when executing SARA on the 10 networks with the number of stations varying from 10 to 100. We investigated

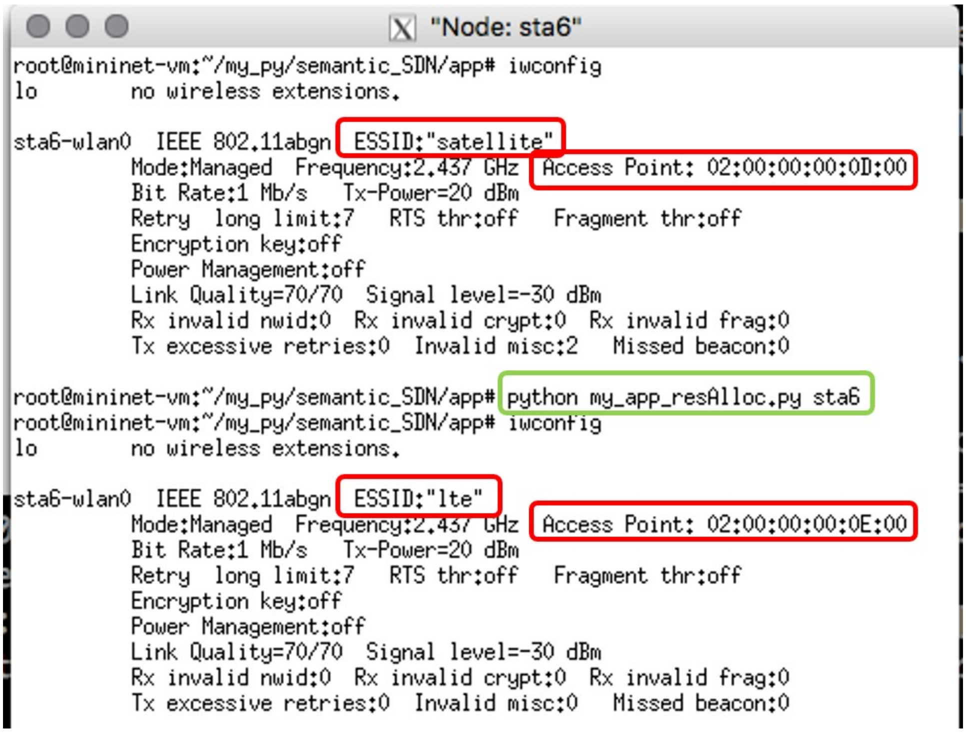


FIGURE 5.2: Screenshot of a command line terminal of a mobile station "sta6", before and after SARA's execution. The "iwconfig" results show that SARA has successfully selected and re-associated to another AP for "sta6".

the relationship between the size of the networks and the size of the knowledge base generated, as shown in Fig. 5.4. An obvious linear positive correlation could be observed from the figure between the scale of the network and the knowledge. The system time, user time, and the total time overhead are calculated and illustrated in Fig. 5.5, Fig. 5.6, and Fig. 5.7, respectively. These time are the time to complete one AP selection cycle: time to 1) scan all available APs, 2) test all available APs, 3) sort the APs and choose the best one. The experiment is repeated 10 times on networks with different number of stations. Each dot represents the system time overhead in one experiment. In all the three figures, Fig. 5.5, Fig. 5.6, and Fig. 5.7, the diameter of a dot represent the size of the knowledge base for each network (the number of RDF triples inside the knowledge base). The time overhead value is denoted as the y-axis, the scale of the network is denoted by the x-axis. As expected, the scale of knowledge base is proportional to the scale of the network. Thus, the diameter of dots grows with the x-axis. It can be shown in Fig. 5.5 that the system time is roughly 0.07s, and there is no obvious increase as the network scales. In some instances, it resulted in the execution time being slightly less than the network with smaller number of nodes,

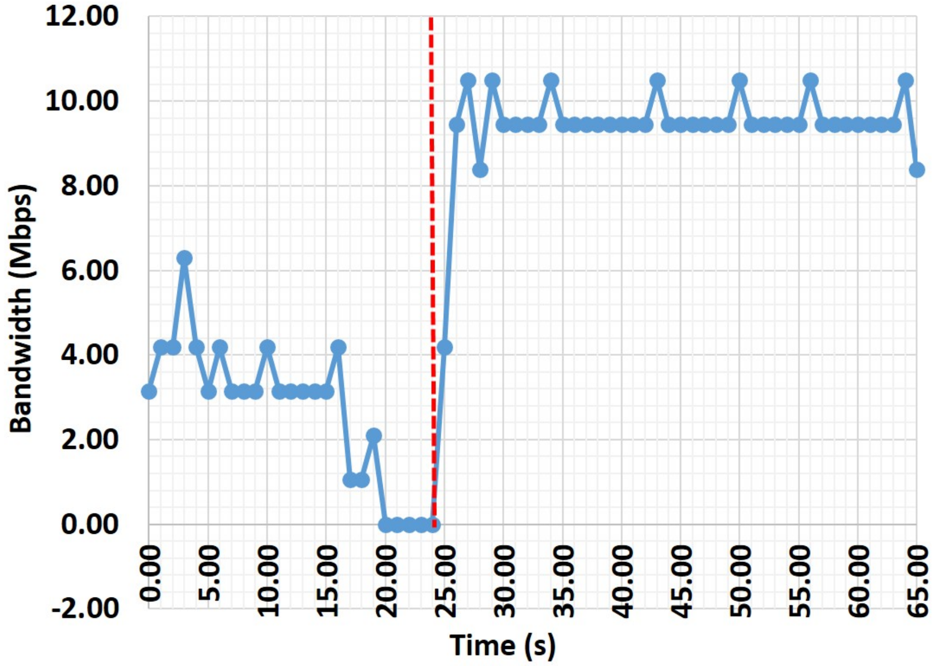


FIGURE 5.3: The bandwidth of the mobile station "sta6" before and after the SARA is executed. The red dash line denotes the time when SARA was executed. As can be seen from the figure, the bandwidth of "sta6" has improved triple than before SARA is executed.

as can be observed from the user time overhead illustrated in Fig. 5.6. Almost all of the experiments consumed 2.7 seconds on user's behalf. The overall time overhead, as shown in Fig. 5.7, stabilised to around 3 seconds.

We argue that even with the above time overhead, SARA is still faster than current existing AP selecting methodologies. Besides, the best AP selected by SARA is more reasonably, considering the real channel situation (in term of bandwidth), the congestion, and the signal strength of each AP as a whole, and it is accomplished in an automatic fashion.

5.4 Conclusions

In current wireless telecommunications networks, a combination of various access technologies is available for mobile users, e.g., WiFi, LTE, 3G/2G, Satellite, LiFi,

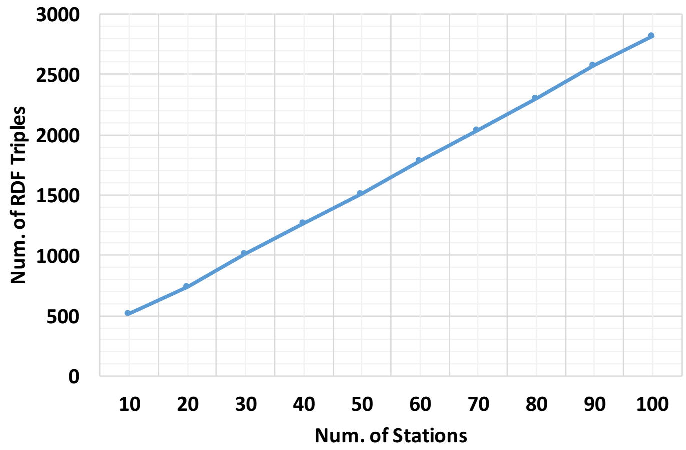


FIGURE 5.4: The scale of the knowledge base, in terms of the number of RDF triples in the knowledge base, for networks with different topologies and scales. An obvious linear positive correlation could be observed from the figure between the scale of the network and the knowledge.

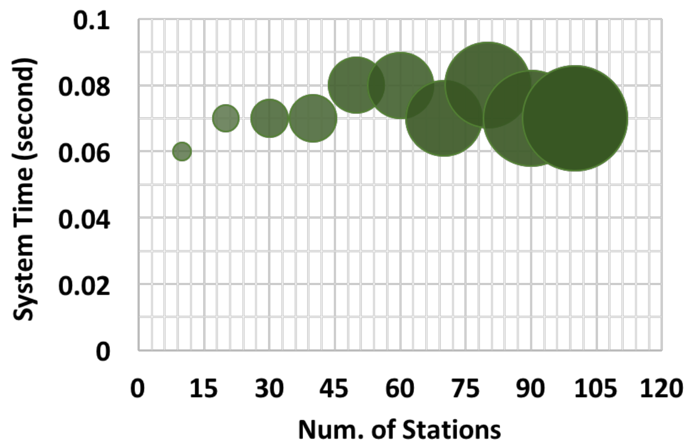


FIGURE 5.5: System time to complete one AP selection cycle: time to 1) scan all available APs, 2) test all available APs, 3) sort the APs and choose the best one. As can be seen from the figure, the system execution time is quite stable with the increase of network scale, at roughly 0.07 *seconds*, which is acceptable for an autonomic AP selection service.

mesh/adhoc. Users need to choose between these technologies to achieve the best communication quality. Current AP selection criteria is based only on signal strength.

We have presented SARA, an autonomic resource allocation service for hybrid wireless networks, as an application of the SEANET system. Based on the knowledge base built by SEANET, SARA can quickly associate with the selected AP taking account of the bandwidth, the congestion, and the signal strength.

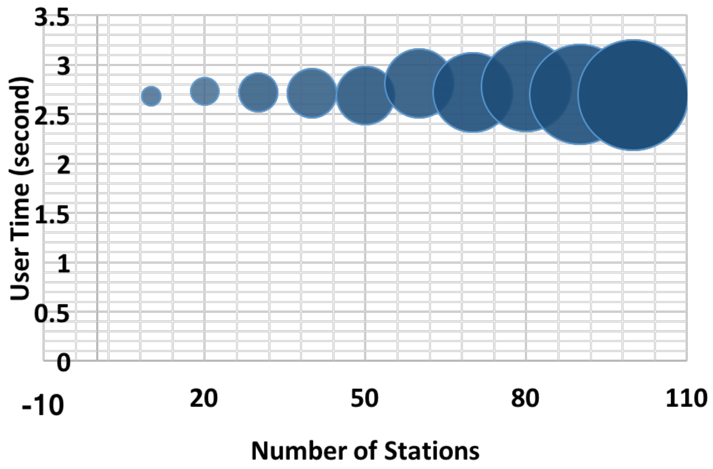


FIGURE 5.6: User time to complete one AP selection cycle: time to 1) scan all available APs, 2) test all available APs, 3) sort the APs and choose the best one.

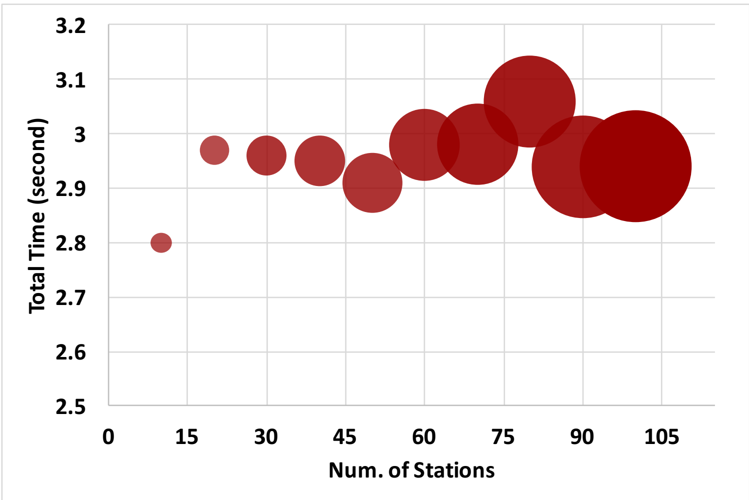


FIGURE 5.7: Total time to complete one AP selection cycle: time to 1) scan all available APs, 2) test all available APs, 3) sort the APs and choose the best one. As can be seen from the figure, the total execution time is quite stable with the increase of network scale, at roughly 3 *seconds*, which is acceptable for an autonomic AP selection service.

We have evaluated SARA in ten hybrid wireless networks of different scales. Our evaluation results proved the accuracy and performance of SARA. Our overhead is acceptable, and our approach autonomic.

Chapter 6

Lost Silence: A shipwreck accident early detection service by continuously process telecommunication data streams

Untimely failures in detection and rescue are the top reasons for large scale fatalities in ship capsizing accidents [185]–[187]. We believe that telecommunication offers a route to successful accident response. In this chapter, a methodology is proposed to detect an accident immediately (with the delay of the order of million-seconds), by processing semantically annotated streams of data in cellular telecommunication systems. With SEANET, the positions and status of phones are encoded as RDF graphs. In this methodology, we propose an algorithm that processes streams of RDF annotated telecommunication data to detect abnormalities. We illustrate our approach with a real shipwreck accident – "Eastern Star" in Hubei China [187]–[189]. Our evaluation results show that with a properly chosen window size, this ship capsizing accident could have been detected. This chapter is based on the work published on [39].

This Chapter is structured as follows. Our motivation are introduced in Section 6.1. Section 6.2 introduces our illustrative scenario based on the "Eastern Star" shipwreck accident. Section 6.3 presents some technical background knowledge of telecommunication networks that are used in our proposal. Section 6.4 illustrates the ontology we have adopted in our proposed methodology. Section 6.5 provides details of the algorithms. The evaluation requirements, results, and discussions based on our scenario are given in Section 6.6. Related work is illustrated in Section 6.7 and conclusions are given in Section 6.8.

6.1 Introduction

At 21:30 UTC+8, on 1st June, 2015, a cruise ship "Eastern Star" traveling on Yangtze River began to tilt due to the stormy weather. Fifty seconds later, the cruise capsized along with 454 people on it. No SOS signal was sent. It was several hours before the security department was aware of this tragedy and 442 lives were lost [189]. In recent times, capsize incidents of refugee boats in the Mediterranean and beyond have been relatively common [190]. In 2016 alone, there are 225,665 refugees get to Europe by sea, and 2,933 lost their lives due to ship capsizes accidents. In 2015, 3,770 refugees lost their lives. And in 2014, that number reached 3,279 [190]. Most of the refugee ships capsizes were undetected until the survivors swam to shore or the shipwreck was spotted by observers. Although sophisticated information and communication technologies (ICTs) have been developed to effectively respond to natural and human-made disasters, the economic cost and fatalities due to untimely rescue show no sign of lessening [188], [190].

When a ship is capsizing and incapable to communicate by radio, is there any way to detect the ship capsizing as soon as it happens? In telecommunication system, the location and status information of user equipment (phone, tablet, wearable devices, etc.) are stored in the service providers' database. This information will show strong patterns when a ship capsizes. More specifically, when a cell phone is normally powered off (e.g., batteries running out, or by pressing the power button), it will register its status as "detached" to the database. However, if a device is shut down forcefully (e.g., physically damaged, dropped in water, or the battery pulled off), or enters a blind zone suddenly which has no signal coverage, it does not have enough time to register its status [191]. Within a location update period (varies from 30 minutes to 1 hour, depending on the service provider), the phone will be marked as "unreachable" in the database, and after the location update, its status will be changed to "detached", which is the same as in a normal shutdown. In a ship capsizing tragedy, a large number of phones will lose signals at the same time. Although there are usually more than one phone operator serving the same geographic location, and the data between different operators is not shared, it is possible that data streams from different phone operators could be queried concurrently. And with the concept of "Geo-Pixel", which is proposed below, the phone status information of the same geographic locations could be easily processed. Thus, by querying in real-time where cell phones lost signal simultaneously, it is possible to detect a ship capsizing tragedy the moment it happens. For example, at the moment when the "Eastern Star" tragedy occurred, more than 400 phone signals were lost in seconds at the same position. Should we be able to query the data streams from telecommunication systems in real time, we could have found this abnormality, and the rescuers could have been sent much earlier. In the refugee vessels on the Mediterranean Sea scenario, it is very likely that the refugees bring their phones with them. Even though the vessels are not equipped with communication systems (or would not use them during illegal migrations), the base stations onshore can still detect the phone signals of the refugees and thus are able to detect any abnormalities.

Lost Silence has been designed to fullfill this task. By semantically integrating heterogeneous data, both real-time traffic data streams and traditional static system records, Lost Silence provide a vision of how the data inside telecommunication system structured by semantic web technologies could help city managers in better decision making.

In the recent few years, RDF stream processing has gained a prominent attention in the Semantic Web community [192]–[195]. It is a technology to reason on the rapidly changing (streaming) data. As the data in real telecommunication system, e.g., SNMP polling, tcpdump, system log, and configuration value, etc., are extremely dynamic (the granularity of SNMP, tcpdump and Syslog update period is in terms of few seconds, microseconds and one second respectively) [196]. Thus, the capability to

Table 6.1: Population and geographic information of the Jianli city.

Town	Area (km^2)	Population
Rongcheng Town	85	$152,358$
Hongcheng Village	254	$133,544$
Water Area	14.520	0

manage rapidly changing system at the semantic level is required. Lost Silence is powered by the Continuous SPARQL (CSPARQL) engine [196]. It is an extension of SPARQL with the ability to query streaming RDF data in real time.

Although the data from mobile phones is increasingly used in social research [197], the early ship disaster detection based on semantically processed telecommunication system data has not yet been proposed to our knowledge. Enormous value of phone information is still hidden in the private databases of telecommunication service providers. (The main reason lies in the fact that current telecommunication systems lack central control, and are not semantically intelligent.) The information model currently adopted cannot support real time queries on the rapid changing data streams at the semantic level.

6.2 Motivation Scenario

We outline a scenario based on the real life cruise ship "Eastern Star" capsizing on the Yangtze River, in Jianli, Hubei, China [189]. In addition to the water area of Yangtze River, the Jianli city consists of two main parts, a densely populated zone – Rongcheng town and a surrounding rural area – Hongcheng village. The detail of population and geographic information is shown in Table 6.1 [198].

In a city with a heavily loaded telecommunication datacenter, with high volumes of phones (approximately 286,000 phones in the city covering an area of 373.520 km^2), we have evaluated the algorithm proposed in this paper against the accuracy of the detection and the time taken to detect abnormalities. In particular, the tradeoffs between the time taken in generating linked-data streams for the massive amount of data in telecommunication system, and the accuracy of the abnormalities detected, have been investigated .

There are possibilities that some situations could reduce the accuracy of the approach described in this paper. For example, although current telecommunication service providers have been trying very hard to achieve a full signal coverage, there are still some regions, especially in rural areas, that have no or limited signal coverage. These areas are known as blind zones (or blind spots). If a large number of phones enter a blind zone, then a large number of phone lost signals at the same location. Each service provider keeps a list of its blind zones. False alarms caused by this could be avoided by a simple comparison to the list before sending an alert.

6.3 Technologies Adopted

6.3.1 Stream processing

In the last few years, RDF stream processing has gained a prominent attention in the Semantic Web community [192], [193], [195], [199]–[202]. The data in real telecommunication system, e.g., tcpdump, system log, SNMP polling, and configuration data, etc., are extremely dynamic. The granularity of Syslog, tcpdump, and SNMP update periods are in terms of microseconds, a few seconds, and one second, respectively [196]. Thus, being able to manage rapidly changing systems at the semantic level will be a significant advantages. The proposed Lost Silence application is powered by the CSPARQL engine [193], an extension of SPARQL with the ability to query streaming RDF data in real time.

The Lost Silence service relies on the up-to-date standard technologies in telecommunication system. In this section, some necessary background knowledge of telecommunication networks based on which Lost Silence is conceived is briefly introduced, along with the technologies we have adopted to implement it.

6.3.2 The location register process in telecommunication networks

To ground our discussion, it is necessary to understand how the telecommunication system maintains records of the location of a phone. In cellular telecommunication networks, phones measure the channel environment and reselect cells every 200ms, and report its location to the cell tower periodically. This period varies from every hour to every 30 minutes, depending on the service providers [197]. When a phone is powered off normally, it will deregister itself to the network management system and its status will be marked as "Detached". However, if a phone abnormally loses its power, it has no time to deregister. Thus its status in the network management system will be "unReachable". After the update period, if the network still does not receive update information from this phone, it will register the phone as "Detached", in the same manner as if it had been shut down normally [191], [197]. Thus, when a phone loses signal abnormally, there is a short time period to identify whether it has been normally shut down or not. Typically this time period is in the order of 30 minutes to an hour.

The phone's location information is sent and stored in network management servers (namely home location register, HLR or visitor location register, VLR) located in the datacenter of the telecommunication system. Normally, there is one datacenter for each service provider per city, managing the data of all the phones in the city. There are usually more than one service provider in a city, and these service providers usually adopt different schemas to represent the same data. In cases when public security is concerned, some information can be shared to an authorised third party. For example, in America, as enforced by the government, the location information from which every 911 call is made is automatically obtained and displayed for the police [203].

A cellular network is a communication network providing wireless connection at the last hop [197]. It covers the geographie area with cells, each one served by an antenna sector of a base station, as shown in Fig. 6.1. The current cellular telecommunication

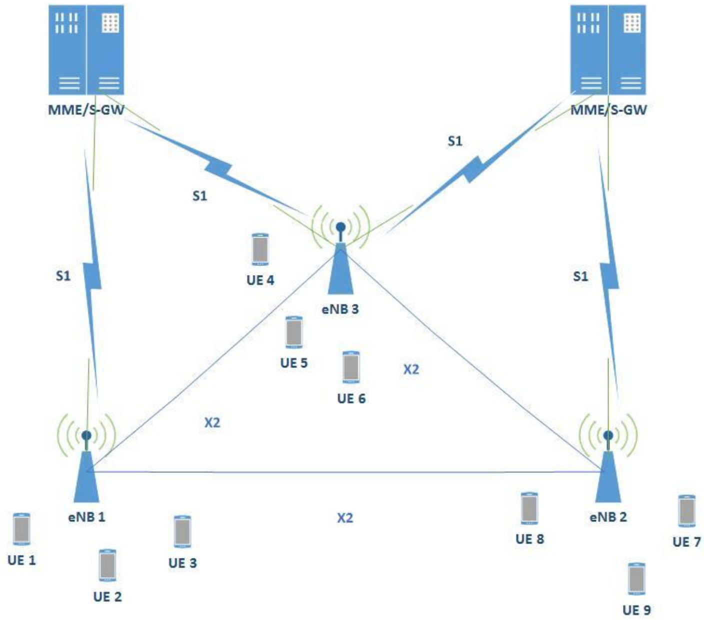


FIGURE 6.1: Architecture of current 4G cellular network.

technology standard is the Long-Term Evolution (LTE), commonly known as the 4th generation (4G) technologies [197].

The LTE technique specifications are tens of thousands of pages now, and they keep updating all the time. A very brief overview of its architecture is given below [197].

- *UE:* short for user equipment, a generalized name for mobile phones, tablet, or any device that could communicate through radio in telecommunication system.

- *eNodeB:* short for Evolved Node B, is the base station towers in LTE. It communicates and controls phones, and directly connects to a core network. ENodeBs are connected with each other, and to the core network through wired interfaces. These interfaces are wired, and reliable, compared to radio access.

- *Core network:* it is responsible for the overall control of the phone and the establishment of the connections. There are many logical nodes in core network, e.g., PDN gateway (P-GW), serving gateway (S-GW), and mobility management entity (MME). P-GW is responsible to allocate IP address for the phone. MME is responsible to process the signaling between the phone and the core network.

The function of Lost Silence depends on the connection termination procedure adopted by the current telecommunication system. When a phone is switched off normally by pressing the power button, a detach indication message will be sent to the network [197]. When the request is accepted, the network will send a detach accepted message to the phone and ask the phone to update its location one last time. After that, the phone is switched off and the connection resource is released. The state of this phone will be registered as "Detached" in the service provider's databases [197]. However, if a phone is forcefully shut off by accident, e.g., drop into water, physically damaged, battery removal, etc., it does not have time to do this requesting procedure. In telecommunication networks, a phone will measure the channel environment and reselect cells every 200ms, and will update its location to the service provider periodically. This period varies from every 30 minutes to every hour, depending on the service providers. Before the update, when somebody tries to call this phone, the network will not be able to locate it, and then will register this phone as "unavailable" in its database [197]. After the update period, if the network still does not receive the update information from this phone, it will register the phone as "Detached", the same as if it is normally shut down. Thus, after a phone lost signal, the only time we have to identify whether it is normally shut down or not, is the time period for location update, about 30 minutes to an hour. Ship capsizing tragedies usually come along with massive damages of phones. A large number of phone losing signal abnormally at the same location can be used as a sign of tragedy if only the telecommunication systems is able to detect it in real-time. Lost Silence is designed to fullfill this task.

During the evolution of wireless telecommunication technologies, the methodologies of location positioning has also evolved from solely satellite based positioning to mobile radio based triangulation positioning and the hybrid approaches [111]. In the current 4G wireless communication systems, two methodologies are provided for the mobile radio based positioning, namely, Observed Time Difference of Arrival (OTDOA) and enhanced Cell ID (eCID), which the positioning accuracy could reach to approximately 1 meter [111].

6.3.3 Methodology

In this section, a detailed description of the Lost Silence service's methodology is given.

6.3.3.1 Geo-Pixel

The key of Lost Silence is to identify the unusual high density of space distribution of phones that lose signals simultaneously. Central to the Lost Silence algorithm is the definition of a standard geographic unit. City pixel, the basic geographic unit used in [204], is the area of a square of $250m \times 250m$. In Lost Silence, we adopt the *geo-pixel*, which is chosen based on the third decimal degree of the latitude and longitude coordination, with the resolution of $0.001° \times 0.001°$, or $100m \times 100m$. The phones that lose their signal are aggregated into different pixels by their rounded up third decimal place of coordination. For example, the phones in an area from a latitude of 15.1905 to 15.1914 and a longitude of 37.2065 to 37.2074 will be all rounded up into the *geo-pixel* (15.191, 37.207), as shown in Fig. 6.2. Thus, the main target of Lost Silence algorithm is to detect the *geo-pixels* with abnormally high number of lost signal.

There is another issue that is worth mentioning. At the equator, the third decimal degree of longitude and latitude both cover about 100 meters. When moving toward the pole, the distance represented by one degree of difference in longitude gets to zero, while the latitude stays almost the same. Because the length of a degree of longitude (east-west distance) depends on the radius of a circle of latitude. For example, one degree of longitude worth up to 111.320 km when latitude is 0, and worth only 28.920 km when latitude is 75°. However, for most countries, the latitude falls between 75° to $-75°$. Thus, the spatial resolution of pixel varies from 100 $m \times 100$ m meter to about 100 $m \times 30$ m meters. Thus, as the size of a common cruise ship varies from 10 meters to almost 100 meters, there are three possible situations for the position of a ship capsizing accident in the grid, namely, inside a pixel, between two pixels, at the intersection of three and four pixels.

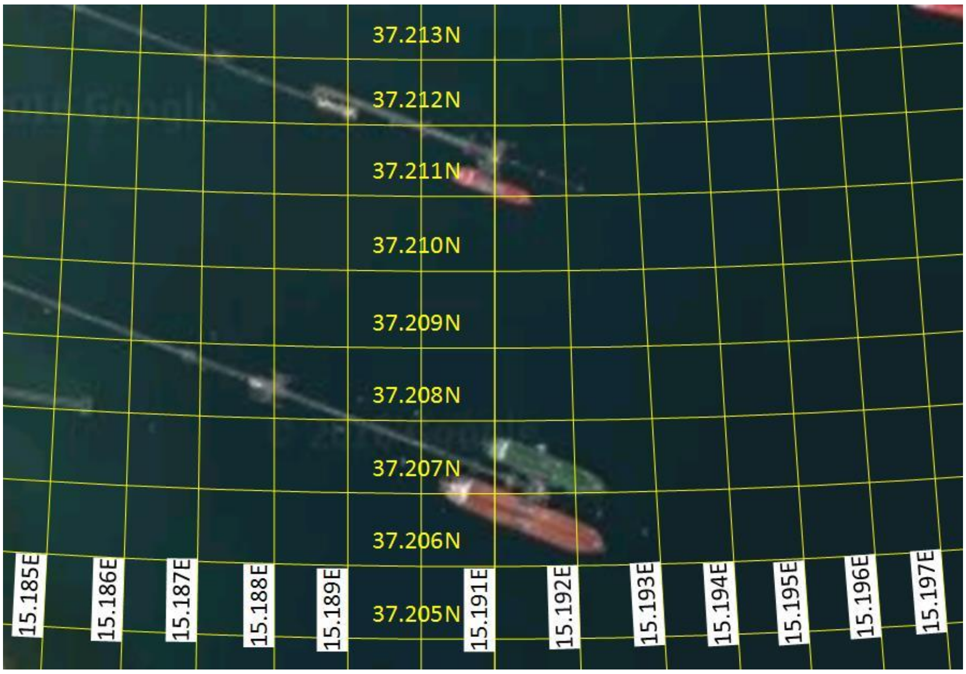


FIGURE 6.2: A schematic of Geo-Pixel

6.3.3.2 Detection

Usually there are many vendors who providing cellular telecommunication service in a single certain geographical area, competing with each other. Most vendors have their own physical infrastructures and the traffic data are isolated and encrypted. In Lost Silence the measurements from phones and eNodeB of different vendors will be merged together and queried by the *geo-pixels* at the data centre. For example, suppose there are three vendors in one area, their live phone data are represented by three ontology streams. If the total number of phones that lost signal in one *geo-pixel* exceed the threshold, and this *geo-pixel* is not a blind zone, then a Lost Silence alert will be sent.

6.4 Ontology adopted in Lost Silence

To be able to transform telecommunication system data into RDF, we adopted ToCo. As stated in Chapter 3, ToCo is proposed to represent the knowledge within a heterogeneous telecommunication system, consisting various technology domains, e.g.,

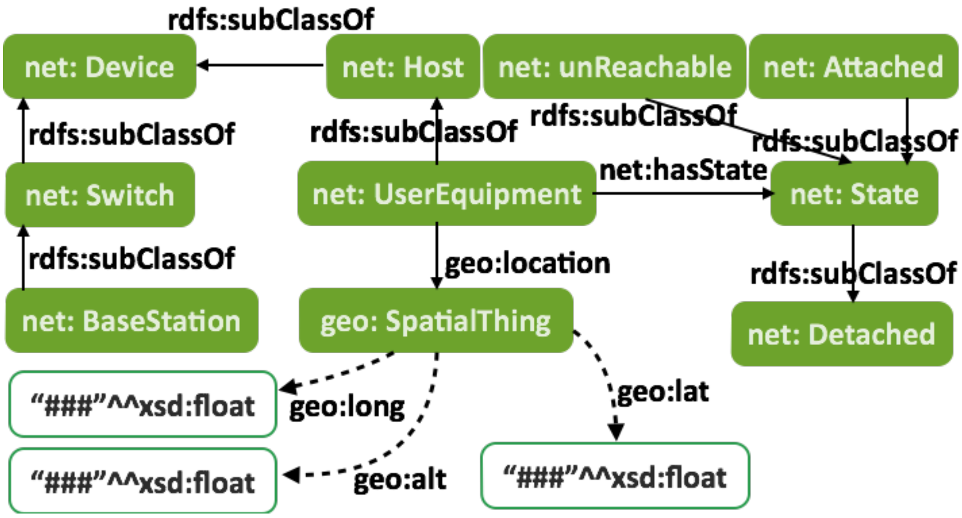


FIGURE 6.3: Classes from the ToCo adopted in the Lost Silence scenario. The class is denoted by solid block, while a hollow block denotes data. The object properties are denoted by solid lines, and dash lines denote datatype properties. The prefixes used in Lost Silence are: *net*: <http://purl.org/toco/>; *geo*: <http://www.w3.org/2003/01/geo/wgs84_pos/>.

mobile network, computer network, optical network, LiFi, and WiFi, etc. A diagram of the portion of the ToCo class hierarchy adopted in Lost Silence is shown in Fig. 6.3.

Some important concepts and relations are presented below.

"UserEquipment" - user equipment in mobile communication system, such as phones, tablets, wearable facilities, etc. It has an object property `hasStatus` with the object of class `Status`. It is defined in the network resource sub-ontology.

"Point" - describes the location of phone, extended from `SpatialThing` of the W3C WGS-84 vocabulary[1]. It is defined in the location sub-ontology. It has three datatype properties, namely, `long`, `lat`, and `alt`, describing the longitude, latitude, and altitude of the phone.

"Status" - the status of UE connectivity in communication system. There are three instances of this class, namely, `Attached`, `Detached`, and `unReachable`, which denotes the phone is connected, detached after deregister to the system, and

[1]See http://www.w3.org/2003/01/geo/ for more detail.

Table 6.2: Phone density of geo-pixels in the city of Jianli

Town	Number of *Geo-pixels*	Phone Density
Rongcheng Town	7024	21
Hongcheng Village	209911	6
Water Area	1200	0

unreachable without deregister to the system, respectively. It is defined in the network resource sub-ontology. It is the range of the property `hasStatus` of `UserEquipment`.

To describe the fact that *"a phone lost signal at location (329.860, 246.792)"*, the corresponding triples are:

```
net:Phone_1 a net:UserEquipment .
pos:Point_1 a pos:Point .
net:Phone_1 pos:location pos:Point_1 .
pos:Point_1 pos:latitude "329.860".
pos:Point_1 pos:longitude "246.792".
net:unReachable a net:Status .
net:Phone_1 net:hasStatus net:unReachable .
```

6.5 Algorithm

As the data volume from telecommunication systems is extremely large, the philosophy of divide-and-conquer is adopted in the algorithm design. In the data centre of the telecommunication system, for our scenario more than 400 phones lost their signal at the same location within seconds[2]. This abnormality could have been detected if continuous queries were being executed on the streams in real-time. An alert could have been raised much earlier and help sent to those on the ship. In the datacenter of the telecommunication system, every time a phone updates its status, a linked-data event is generated and published as a collection of triples. As the phones in the city keep updating their location information and signal status, a linked-data stream is generated. In order to estimate the volume and velocity of data variation from

[2] By personal correspondence.

telecommunication system during accidents, we referred to the literature. The mobile phone penetration rate is 96.7% in China, [198]. With the *geo-pixel* simplification, the number of geo-pixels and the population density of phones in each town are shown in Table 6.2. As a result, in the city of Jianli, there are 29,215 geo-pixels, in which 7,024 of them are in the densely populated zone with on average 21 phones per pixel, 20,991 pixels are in rural areas with on average 6 phones per pixel, and on the river there are 1,200 pixels where the phone density is on average 0 phones per pixel.

```
REGISTER QUERY StreamingAndExternalStaticRdfGraph AS
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pos: <http://www.w3.org/2003/01/geo/wgs84_pos/>
PREFIX net: <http://purl.org/toco/>
PREFIX fn: <http://www.w3.org/2005/xpath-functions#>
SELECT (COUNT(?UE) AS ?counter) ?lat ?long
FROM STREAM <http://purl.org/toco/stream> [RANGE 30m STEP 5s]
WHERE {
        ?UE net:hasStatus net:unReachable.
        ?UE pos:location ?point.
        ?point pos:lat ?lat;
                pos:long ?long.
        BIND (fn:round(?lat * 1000) as ?roundLat)
        BIND (fn:round(?long * 1000) as ?roundLong)
 }
GROUP BY ?roundLat ?roundLong HAVING (?counter >10)
```

Listing 6.1: C-SPARQL query string for detecting the lost phones. Each non-empty result in the returned result sets denotes an alert.

// Input are the longitude and latitude of the city, ratio of phone lost signal, thread sleep time. Output is the RDF stream created
Require: ***lat, long, lostRatio, sleepTime***
Ensure: ***Stream***
keepRunning $\Leftarrow 1$
while keepRunning **do**
Insert triple (net:Phone, geo:locateIn, geo:Point) to ***Stream***
Insert triple (geo:Point, geo:latitude, "***lat***"^^xsd:double) to ***Stream***
Insert triple (geo:Point, geo:longitude, "***long***"^^xsd:double) to ***Stream***
With a ratio of ***lostRatio***:
Insert triple (net:Phone, net:hasStatus, net:unReunReachableachable) to ***Stream***
With a ratio of(1 − ***lostRatio***):
Insert triple (net:Phone, net:hasStatus, net:Attached) to ***Stream***
Thread sleep for ***sleepTime*** seconds
end while

Algorithm 4: Stream Generator: Generate RDF streams for each given *geo-pixels*.

The C-SPARQL query string to detect lost phones is shown in Listing 6.1, the .sparql file could be found in Appendix C.2. One C-SPARQL instance is generated for each

// Input is the number of *geo-pixels* for city center, rural area, and water zone. Output is the tragedy detection results of the city
Require: ***CityNum, RuralNum, WaterNum***
Ensure: ***Gstream***
Instantiate a blocking queue: ***q***
Instantiate a thread pool: ***pool***
for each of the ***CityNum / RuralNum / WaterNum*** *geo-pixels* in the **City / Rural area / Water zone do**
    Create an **Stream Generator** as a thread into ***pool***
    Apply C-SPARQL query on the stream
end for
Shut down ***pool***

ALGORITHM 5: Thread Pool Scheduler: Schedule the C-SPARQL engine threads.

geo-pixel. Thus, there are 29,215 C-SPARQL instances running in the experiment. With the massive data volume in telecommunication system and the heterogenous population density, it is unrealistic to execute the query based on the arrival of new triples as CQELS does [200]–[202]. Thus, we choose C-SPARQL, which supports a time step execution model. The query is evaluated every 5 seconds calculating for each *geo-pixel* the count of unreachable devices over the last 30 minutes. A result is only returned if the count in the *geo-pixel* is above 10. Each stream is denoted as <`http://purl.org/toco/stream`> in the query. To avoid the omissions of accidents, we choose the window size the same as the phone's position update period in communication system – 30 minutes. We adopt a *thread-per-geo-pixel* architecture to process telecommunication data and generate an RDF stream per *geo-pixel* within one thread. The threads are maintained by a thread pool. Each thread in the pool has the same execution priority and receives an equal share of CPU time.

The details of the generated streams and thread scheduling process are illustrated in a self-explanatory way in Algorithm 4 and 5 for brevity. Algorithm 4 shows the stream generating process for different areas. The *lostRatio* is adopted to control the ratio of phones that lose signal in the simulated streams. For example, if the *lostRatio* = 0.1, it means 10% of the phones in this stream will lose signal, and if the *lostRatio* = 1, it denotes all the phones in this stream will lose signal. The density of phones is determined by the *sleepTime*. For example, in the rural area Hongcheng Village where the phone density is 6 phones per pixel, as the location update period is 5 *seconds* (= 5000 *milliseconds*), in order to simulate the scenario that there are 6 phones signal update per 5 *seconds*, we have to generate the RDF triples for one phone every $5000 \div 6 \approx 833.333$ *milliseconds*. Thus, the *sleepTime* for rural area is 833 *milliseconds*. Similarly, the *sleepTime* for Rongcheng Town is $5000 \div 21 \approx 238$ *milliseconds*.

6.6 Evaluation

We run an extensive and exhaustive evaluation based on the tragedy detection algorithm. The evaluation was carried out on a MacBook Air OS X 10.9.5 with 3MB cache running on Intel Core i5 at 1.5 GHz. The system has 128 GB SSD and 4 GB RAM.

6.6.1 Preparing the Data

A small number of phones will randomly disappear from random pixels along the river. At a randomly chosen time, 424 phones will lose their signal simultaneously in the geo-pixel (329.863, 246.792). With a query window of 30 minutes, we experimented with query steps of 5, 20, and 30 seconds respectively. For the three step sizes, we ran in total 3 iterations of the algorithm. Our evaluation for the shipwreck scenario focuses on two criteria: the time taken to detect abnormalities and the accuracy of the detection, in terms of the cases of fail-to-report.

6.6.2 Experimental Framework

Fig. 6.4 shows our workflow and execution environment of the Lost Silence experiment. Data from the telecommunication system in each *geo-pixel* is converted into RDF streams by the Stream generator. RDF streams are accessed and queried by the C-SPARQL engine thread. All the 29,215 query threads processing 285,902 phone data are scheduled and executed by thread pool.

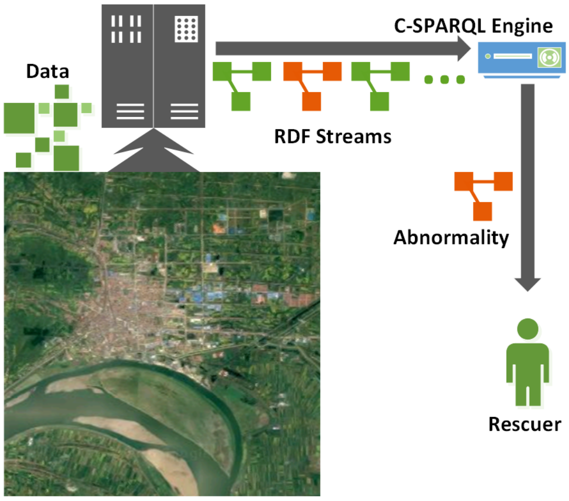
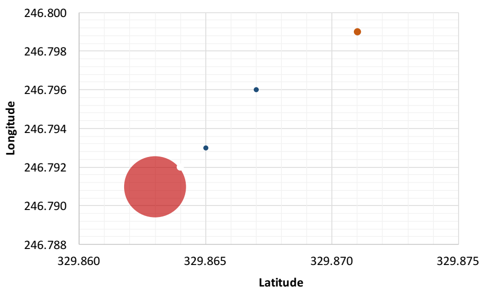


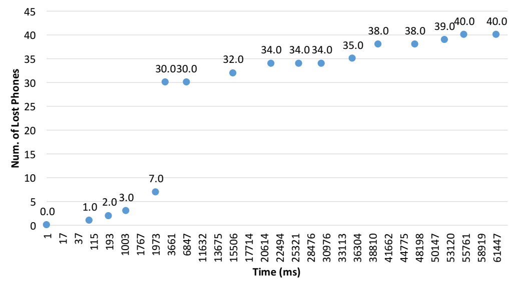
FIGURE 6.4: The process of disaster early detection in Lost Silence.

6.6.3 Evaluation Results

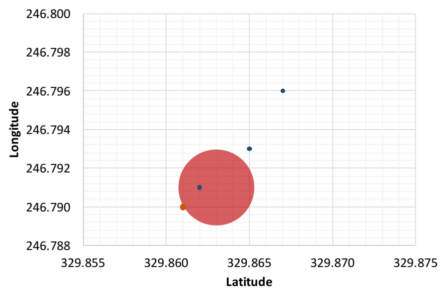
We run the experiment three times, with the query steps of 5 seconds, 20 seconds, and 30 seconds, respectively, and the window size is 30 minutes. For example, when the query step is 5 seconds, it means the query is executed every 5 seconds on the streaming data from the start of the query. Only the data streams from current back to 30 minutes before is queried, any data more than 30 minutes ago is not cared and gone with the time. The simulation results are shown in Fig. 6.5. The Fig. 6.5(a) and Fig. 6.5(b) are the simulation result with the query step of 5 seconds, the Fig. 6.5(c) and Fig. 6.5(d) are with the query step of 20 seconds, and the Fig. 6.5(e) and Fig. 6.5(f) 30 seconds. In the Fig. 6.5(a) Fig. 6.5(c), and Fig. 6.5(e) on the left, the horizontal and vertical coordinates indicate the latitude and longitude of the geographical area respectively. The total number of lost phones detected at each *geo-pixel* are shown as circles on the subfigures. The number of lost phones are denoted as the radius and colour of the circles, e.g., the smallest circle in black denotes only one lost phone at that location, the second smallest circle in orange denotes two lost phones, and the largest circle in red is the total 442 lost phones detected where the shipwreck took place. The Fig. 6.5(b), Fig. 6.5(d), and 6.5(f) on the right show the number of lost phones detected in the geo-pixel (329.863, 246.792), where the shipwreck took place, at each query step, from the beginning of the experiment. To shed light on



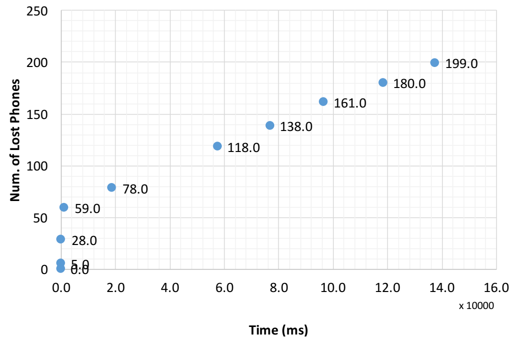
(a)



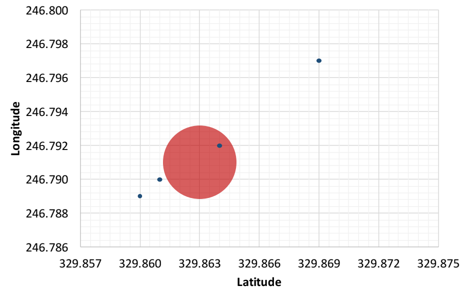
(b)



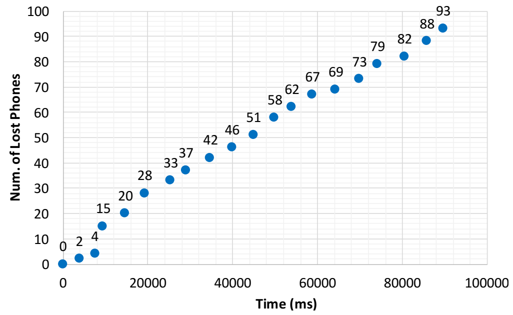
(c)



(d)



(e)



(f)

Figure 6.5: The total number of lost phones in the shipwreck zone and the CSPARQL query results in the *geo-pixel* (329.863, 246.792), in which the shipwreck took place, at each step for the three experiments with the query steps of 5 seconds, 20 seconds, and 30 seconds, respectively. Fig. a) and b) illustrate the experiment with the query step of 5 seconds, Fig. c) and d) illustrate 20 seconds, and Fig. e) and f) 30 seconds. Fig. a), c), and e) present the total number of phones that lost signal in the water area of city Jianli at each experiment. The number of phones lost signal is illustrated with the colour and radius of the circle, e.g., the brighter colour and larger radius of a circle denotes a larger number of lost phones. Fig. b), d), and f) show the detected number of phones that lost signal in the *geo-pixel* (329.863, 246.792) from the beginning of the query, at each query step .

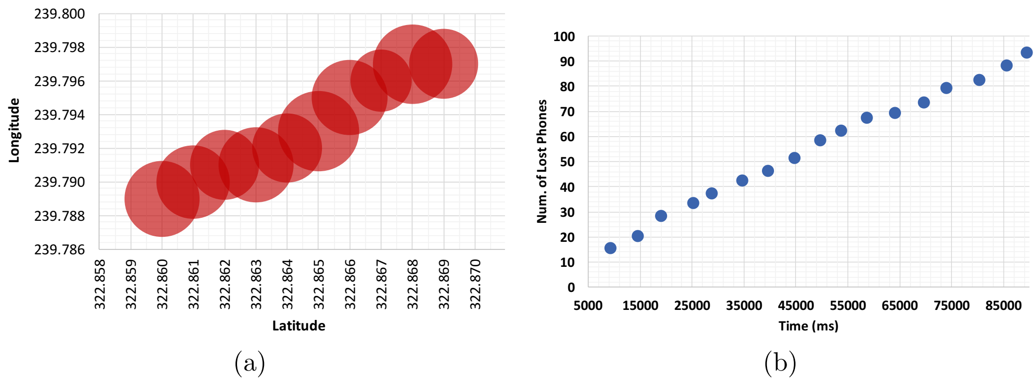


FIGURE 6.6: Query result of the pressure test, in which massive phones lost signal in multiple *geo-pixels* along the river.

the detailed detection process, we show the number of lost phones below 10 as well, although no alert will be send for that. As shown in the Fig. 6.5(b), Fig. 6.5(d), and Fig. 6.5(f), there is a surge in the number of lost phones in each figures, at the time when the shipwreck happens. Once the number of lost phones raised to above 10, the Lost Silence will send an alert that abnormally surge with the information of the geo-pixel. As the query step increases, from 5 seconds (as shown in Fig. 6.5(b)) to 30 seconds (as shown in Fig. 6.5(f)), the number of lost phones experiences faster increase rate, and the slopes of the dots become steeper.

A pressure test is also carried out to simulate an extreme scenario in which 10 ships capsize in different locations on the river simultaneously, as shown in Fig. 6.6. Each ship carries more than 300 passengers. The duration of the capsizing of one ship is about 3 minutes. Thus, in the 10 *geo-pixels* along the river, there is a massive phone loss incident with the rate of about $300 \div 3 = 100$ phones per minute. As shown in the Fig. 6.6(a), the abnormality of a large number of lost phones in all of the 10 *geo-pixels* are detected, and alerts are sent successfully. The results at each query step for an random *geo-pixel* is shown in the Fig. 6.6(b). A sharp increase of the lost phones can be spotted, with the rate about 100 phones per minute. The result demonstrates the efficiency and practicality of the Lost Silence.

6.7 Related Work

The utility of data from telecommunication system in geography and social science to improve urban planning has been increasingly investigated [191], [204]–[207]. In [191], Willessan presented how evidence obtained from mobile system plays a part in forensic investigation. [204]–[207] provided a extensive coverage of the smart city applications adopting data from telecommunication system. However, these applications mainly focus on mobile positioning only. [208] presented an interesting application adopting satellite images and linked geographic data to detect wildfire. Ontology and RDF stream processing had also been adopted to develop autonomous vehicles in [209]. A novel approach for spatiotemporal query linked data was reported in [210].

Scholte and Rozenkrane [211] have proposed a system to localize and track each ship, and send personalized alert to those that are expected to be in danger. However, this system might lose function when the communication device fails, and cannot detect an accident when the communication device on the ship is not functioning. [212] designed and developed an ontology for emergency notification, such as "a typhoon approaching", but not for detection tragedies that have already happened.

6.8 Conclusions

In this chapter we have illustrated the power of semantic stream processing by presenting a ship capsizing accident detection service "Lost Silence" based on telecommunication system data. We have shown that the system data inside cellular networks can be used to detect accidents when a large number of phones lose signal at the same location simultaneously, in our scenario a ship capsizing incident.

In Lost Silence, heterogenous user data from all the vendors are concentrated and organised with Geo-Pixel, which is proposed as a geographical unit. Continuous RDF processing is adopted to query the data streams in real time. We perform the simulation based on a real life shipwreck incident in China 2015, and discussed the results in different scenarios.

This Lost Silence service adopts the ontology developed and built for the TOUCAN project. Current ontologies of telecommunication networks generally do not provide information and knowledge inside the system. In our future work, we intend to extend Lost Silence with machine learning approaches and real data from cellular networks.

Chapter 7

Conclusions and Future Work

The motivation for this thesis was to make the nodes inside hybrid telecommunication networks, even if they are from different technology domains, to efficiently exchange messages. Thus, a common language interpretable by machines is required for telecommunication networks.

The previous chapters have developed a formal ontology describing hybrid telecommunication networks (Chapter 3), an autonomic network management system for creating and inferencing knowledge bases automatically, a technology-independent API to publish the functions of the autonomic network management system (Chapter 4). This system has been evaluated by use cases of the system in range of both network management (Chapter 5) and disaster response applications (Chapter 6). This chapter will briefly summarise the main results of the work and suggest future work.

This chapter concludes this thesis in the following structure. Section 7.1 reviews the three questions raised at the beginning, which unfold the story line of this thesis. Based on these questions, the thesis is organised into three parts. Later in Section 7.1 we discuss how each of these questions are answered by the chapters of this thesis. Section 7.2 raises two interesting future research directions, structured machine learning for SDN management, and RDF stream processing on telecommunication network data.

7.1 Conclusions and Results

The goal of the thesis was to answer the three questions raised in the beginning of the thesis (Section 1.1), repeated as below.

Q1. *Is it possible to build an information model describing all of the information of the heterogeneous telecommuication networks, which is universally accepted by all the technology domains and is machine interpretable?*

Q2. *Can we build a knowledge-based network management system?*

Q3. *What applications can be developed based on it?*

To answer these questions, the main part of the thesis was organised into three parts: Chapter 3 answers question Q1; Chapter 4 provides a solution to question Q2; the third part consists of Chapter 5 and Chapter 6 which demonstrate some possible answers for question Q3.

In the first part of the thesis we have investigated the knowledge presentation for telecommunication networks, specifically the SDN, which is believed to be the future of telecommunication networks. After reviewing recent research, we came to the conclusion that an ontology driven knowledge base provides a promising solution for a semantic-intelligent, machine interpretable network information modelling. Knowledge based network management taking advantage of SDN is the only solution for an autonomic network management system. In Chapter 3, we constructed a formal ontology – ToCo[1] for the telecommunication networks with hybrid technologies. The examples and use cases provided demonstrate the practicality of ToCo.

Chapter 4 of the second part of the thesis presents our solution for a knowledge-based autonomic network management system for SDN – SEANET. It is a lightweight, technology-independent agent for network management. SEANET can construct a formally defined ontological knowledge base augmented by inference rules. The machine-understandable knowledge base allows end users to collect, process, inference,

[1] http://purl.org/toco/

and operate on telecommunication networks and the information beneath them, in an autonomic fashion. With the open network management API provided, SEANET allows researchers to develop their semantic-empowered applications in their own production networks. The demonstrations and evaluation results provided on different network topologies and scales, have proven that our system is capable to carry out many network management tasks automatically, e.g., connecting one or more hosts to a switch, adding one or more flow entries to a flow table, or building a firewall, saving human beings from the tedious job of reconfiguring the network step by step.

The third part of the thesis provides more use cases adopting the ontology and knowledge based system SEANET built in Chapter 3 and Chapter 4. SARA, presented in Chapter 5, is an autonomic resource allocation service in hybrid wireless networks, which is able to select and associate to the best AP among APs with different wireless access technologies. It is proposed to replace the current AP selection strategy, which depends on human interven and selecting the AP based on signal strength only. The existing AP selection strategy is not reasonable and does not reflect the real channel condition. Based on knowledge base built by SEANET, SARA is able to quickly identify an AP considering the bandwidth, the congestion, and the signal strength. The evaluation results have proven the accuracy and practicability of SARA, which can increase the bandwidth by three times, with less time overhead.

Chapter 6 proposed Lost Silence, an algorithm that processed streams of RDF annotated telecommunication data to detect abnormality. By encoding the information inside telecommunication networks, such as the position and status of phones, as streaming RDF graphs, the ship capsizing accidents can be detected effectively. The exemplification of our approach is by simulating a real shipwreck accident – "Eastern Star" in Hubei, China. The results have demonstrated the efficiency and practicality of the Lost Silence approach.

7.2 Future Work

Although this thesis has proposed a solution for ontology driven knowledge based network autonomic management system, it is far from a complete application for this area. Thus, the work of the thesis gives rise to a number of interesting future research directions, as stated below.

7.2.1 Structured Machine Learning for Autonomic Network Management System

Telecommunication management system collect massive amounts of data from network equipments and users, providing a big dataset about the network and customers. How to recognise the patterns within the big data and achieve autonomic network management with machine learning has been heatedly discussed [112], [213]–[217]. Machine learning, especially deep learning, has been adopted in various domains of network management, such as cognitive radio, traffic predicting, energy efficiency [112], [213], [214]. The mission of machine learning is to bring forth the potentially valuable information hidden in raw datasets [218]–[223]. In reality, the raw data is often disappointing of low quality [224]–[226]. Machine learners need to comb through hundreds of thousands of datasets, tracking down missed values, correcting format errors, etc. A survey on data scientists conducted by Google [227] shows that the most time-consuming issue during work is the cleaning of dirty data. The knowledge base constructed by ontology offers a promise to facilitate data scientists of formally structured, semantic intelligent datasets. Recently, machine learning on structured knowledge base modelled with RDF, OWL, or other semantic knowledge presentation language has received heated discussion [218], [228]–[232]. This technology has been pinned down as *structured machine learning* by Westphal, et al. [232]. There are three main benefits of the structured machine learning.

- The datasets are formally structured and ready to learn.

- There could be complex knowledge and rules in the datasets.

- As the datasets is modelled by semantic web technologies, the results are interpretable by both machine and humans.

7.2.2 RDF Stream Processing on Telecommunication network data

Telecommunication networks are dynamic. The data gathered in the network is updated dynamically, with various periods. Current analysis on network data models can only been performed on historical data, i.e., data collected and stored on a server. However, questions like "will there be a traffic jam on link_1?", "how many pieces of user equipment are losing signal now?", "Find the top 5 users that generate the highest traffic for the past 15 seconds.", cannot be answered with current network analysing tools. Although dynamic data can be analysis on-the-fly by data stream management systems, but current systems are unable to perform complex reasoning [192]. Although Lost Silence in Chapter 6 has demonstrated a use case of RDF stream processing in telecommunication network, it is based on emulated telecommunication networks only. One interesting future direction would be developing an stream knowledge based network management system, which supports the following functions:

- A stream generator to generate a RDF streams from the ontological structured data inside the network.

- A stream SPARQL engine support reasoning over streaming data.

7.2.3 Ontological Defined Network in Lancaster

The work proposed in this thesis has been implemented in Lancaster University [169]. ToCo has been adopted in their approach toward an ontological defined network. They will continue with this work using semantic web technologies to enrich their data models with reasoning rules and integrity/consistency constraints, and automate state inference across layers.

Appendix A

The Example Competency Questions for TOUCAN Ontology (ToCo)

In a cellular network:

CQ1.1 What is total data traffic in the city centre/rural areas in City X?

CQ1.2 What is the range/number of clients of the base station tower?

In a computer network:

CQ2.1 Do the hosts connect to switch 1?

CQ2.2 Which ports are they connecting to?

CQ2.3 What are their IP and MAC addresses?

CQ2.4 Are they working now?

In a WiFi network:

CQ3.1 What is the signal strength of an access point in a WiFi network?

CQ3.2 How many clients is it serving?

CQ3.3 Are these clients moving?

CQ3.4 What are their velocity?

For applications:

CQ5.1 What content are students viewing on their phone in school/university?

CQ5.2 How many phone signals were lost in a certain area?

CQ5.3 What was their last available position?

Appendix B

The TOUCAN Ontology (ToCo)

```
@prefix : <http://www.w3.org/2002/07/owl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos/> .
@prefix om: <http://purl.oclc.org/net/unis/ontology/sensordata.owl/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix toco: <http://purl.org/toco/> .
@prefix toco2: <https://w3id.org/toco/> .
@prefix units: <https://sweet.jpl.nasa.gov/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://purl.obolibrary.org/obo/UO_0000008> a :Class ;
    rdfs:label "meter" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000001> .

<http://purl.obolibrary.org/obo/UO_0000010> a :Class ;
    rdfs:label "second" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000016> a :Class ;
    rdfs:label "millimeter" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000001> .

<http://purl.obolibrary.org/obo/UO_0000017> a :Class ;
    rdfs:label "micrometer" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
```

```
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000001> .

<http://purl.obolibrary.org/obo/UO_0000028> a :Class ;
    rdfs:label "millisecond" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000031> a :Class ;
    rdfs:label "minute" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000032> a :Class ;
    rdfs:label "hour" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000034> a :Class ;
    rdfs:label "week" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000035> a :Class ;
    rdfs:label "month" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000036> a :Class ;
    rdfs:label "year" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000003> .

<http://purl.obolibrary.org/obo/UO_0000092> a :Class ;
    rdfs:label "turns per second" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000058> .

<http://purl.obolibrary.org/obo/UO_0000094> a :Class ;
    rdfs:label "meter per second" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000060> .

<http://purl.obolibrary.org/obo/UO_0000106> a :Class ;
    rdfs:label "hertz" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000105> .
```

```
<http://purl.obolibrary.org/obo/UO_0000114> a :Class ;
    rdfs:label "watt" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000113> .

<http://purl.obolibrary.org/obo/UO_0000223> a :Class ;
    rdfs:label "watt-hour" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000111> .

<http://purl.obolibrary.org/obo/UO_0000224> a :Class ;
    rdfs:label "kilowatt-hour" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000111> .

<http://purl.obolibrary.org/obo/UO_0000232> a :Class ;
    rdfs:label "bit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000231> .

<http://purl.obolibrary.org/obo/UO_0000233> a :Class ;
    rdfs:label "byte" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000231> .

<http://purl.obolibrary.org/obo/UO_0000235> a :Class ;
    rdfs:label "megabyte" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000231> .

<http://purl.obolibrary.org/obo/UO_0000325> a :Class ;
    rdfs:label "megaHertz" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000105> .

<http://purl.obolibrary.org/obo/UO_0000331> a :Class ;
    rdfs:label "gigabyte" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000231> .

<http://purl.obolibrary.org/obo/UO_0000332> a :Class ;
    rdfs:label "terabyte" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000231> .

<http://purl.obolibrary.org/obo/UO_0010005> a :Class ;
    rdfs:label "millimeters per day" ;
```

```
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000060> .

<http://purl.obolibrary.org/obo/UO_0010008> a :Class ;
    rdfs:label "kilometer per hour" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000060> .

om:hasUnit a :DatatypeProperty ;
    rdfs:comment "reused from SENSEI Observation and Measurement Ontology."^^xsd:
        string ;
    rdfs:domain om:ObervationAndMeasurement ;
    rdfs:isDefinedBy om: ;
    rdfs:range xsd:string .

om:hasUnits a :ObjectProperty ;
    rdfs:comment "reused from SenSei Observation and Measurement Ontology (http://
        purl.oclc.org/net/unis/ontology/sensordata.owl/)."^^xsd:string ;
    rdfs:domain om:ObervationAndMeasurement ;
    rdfs:isDefinedBy om: ;
    rdfs:range <http://purl.obolibrary.org/obo/uo.owl/unit> .

om:hasValue a :DatatypeProperty ;
    rdfs:comment "reused from SENSEI observation and measurement ontology."^^xsd:
        string ;
    rdfs:domain om:ObervationAndMeasurement ;
    rdfs:isDefinedBy om: ;
    rdfs:range xsd:float .

toco:APsInRange a :ObjectProperty ;
    rdfs:domain toco:UserDevice ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:AccessPoint .

toco:Attached a :Class ;
    rdfs:subClassOf toco:UserDeviceStatus .

toco:AudioService a :Class ;
    rdfs:subClassOf toco:Service .

toco:AudioSession a :Class ;
    rdfs:subClassOf toco:Session .

toco:BaseStation a :Class ;
    rdfs:comment """a "land station in the land mobile service." -- according to the
        International Telecommunication Union's (ITU) Radio Regulations (RR).
```

```
it is a transceiver connecting a number of other devices to one another and/or to a
    wider area. In LTE, it is also known as eNodeB."""^^xsd:string ;
    rdfs:subClassOf toco:SystemDevice .

toco:CellularUserEquipment a :Class ;
    rdfs:comment "User deivce in a Cellular network."^^xsd:string ;
    rdfs:subClassOf toco:UserEquipment .

toco:Channel a :DatatypeProperty ;
    rdfs:comment "the channel id of a interface."^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:int .

toco:Detached a :Class ;
    rdfs:subClassOf toco:UserDeviceStatus .

toco:Drop a :Class ;
    rdfs:subClassOf toco:Action .

toco:FileTransferService a :Class ;
    rdfs:subClassOf toco:Service .

toco:Forward a :Class ;
    rdfs:subClassOf toco:Action .

toco:Frequency a :DatatypeProperty ;
    rdfs:comment "the frequency of the radio wave transmitted by a wireless interface
        (WLAN)."^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:range xsd:float .

toco:From a :ObjectProperty ;
    rdfs:comment "the source node that the link is transmit from."^^xsd:string ;
    rdfs:domain toco:Link ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Interface ;
    rdfs:subPropertyOf :topObjectProperty .

toco:Host a :Class ;
    rdfs:comment "A host is a computer or other device connected to a computer
        network. It is a network node that is assigned a network address."^^xsd:
        string ;
    rdfs:subClassOf toco:UserDevice .

toco:Output a :Class ;
    rdfs:subClassOf toco:Action .
```

```
toco:PathFlow a :Class ;
    rdfs:subClassOf toco:Flow .

toco:SatelliteGroundStation a :Class ;
    rdfs:comment "used in satellite network. Also known as earth station, or earth
        terminal, it is a terrestrial radio station designed for extraplanetary
        telecommunication with spacecraft (constituting part of the ground segment of
         the spacecraft system), or reception of radio waves from astronomical radio
        sources. Ground stations may be located either on the surface of the Earth,
        or in its atmosphere.[1] Earth stations communicate with spacecraft by
        transmitting and receiving radio waves in the super high frequency or
        extremely high frequency bands (e.g., microwaves). When a ground station
        successfully transmits radio waves to a spacecraft (or vice versa), it
        establishes a telecommunications link. A principal telecommunications device
        of the ground station is the parabolic antenna."^^xsd:string ;
    rdfs:subClassOf toco:UserEquipment .

toco:SatelliteRelayStation a :Class ;
    rdfs:comment "An artificial satellite, that relays and amplifies radio
        telecommunications signals via a transponder; it creates a communication
        channel between a source transmitter and a receiver at different locations on
         Earth. Communications satellites are used for television, telephone, radio,
        internet, and military applications."^^xsd:string ;
    rdfs:subClassOf toco:SystemDevice .

toco:Switch a :Class ;
    rdfs:comment "A network switch (also called switching hub, bridging hub,
        officially MAC bridge) is a computer networking device that connects devices
        together on a computer network by using packet switching to receive, process,
         and forward data to the destination device."^^xsd:string ;
    rdfs:subClassOf toco:SystemDevice .

toco:Text a :Class ;
    rdfs:subClassOf toco:Service .

toco:To a :ObjectProperty ;
    rdfs:comment "the destination node of the link."^^xsd:string ;
    rdfs:domain toco:Link ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Interface ;
    rdfs:subPropertyOf :topObjectProperty .

toco:VideoService a :Class ;
    rdfs:subClassOf toco:Service .

toco:VideoSession a :Class ;
```

```
    rdfs:subClassOf toco:Session .

toco:WiFiAccessPoint a :Class ;
    rdfs:comment "An access point in a WiFi network"^^xsd:string ;
    rdfs:subClassOf toco:AccessPoint .

toco:WiFiUserEquipment a :Class ;
    rdfs:comment "a user quipment in Wi-Fi network"^^xsd:string ;
    rdfs:subClassOf toco:UserEquipment .

toco:contributor a :AnnotationProperty .

toco:cookie a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:driver a :DatatypeProperty ;
    rdfs:domain toco:AccessPoint ;
    rdfs:range xsd:string .

toco:flags a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:int ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:hard_timeout a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:hasAntennaGain a :DatatypeProperty ;
    rdfs:comment "the gain of an antenna on the wireless interface."^^xsd:string ;
    rdfs:domain toco:WLAN ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:int .

toco:hasAntennaHeight a :DatatypeProperty ;
    rdfs:comment "the height of the antenna of a wireless interface"^^xsd:string ;
    rdfs:domain toco:WLAN ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float .

toco:hasAssociatedStations a :ObjectProperty ;
```

```
    rdfs:comment "in wireless networks, the user equipments (mobile stations in
        another term, e.g., phones, laptops, tablets, etc.) that are associated to
        the access point."^^xsd:string ;
    rdfs:domain toco:AccessPoint ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:UserDevice ;
    rdfs:subPropertyOf :topObjectProperty .

toco:hasBandwidth a :ObjectProperty ;
    rdfs:comment "the bandwidth of the link"^^xsd:string ;
    rdfs:domain toco:Link ;
    rdfs:range om:ObervationAndMeasurement ;
    rdfs:subPropertyOf toco:hasLinkProperty .

toco:hasDevice a :ObjectProperty ;
    rdfs:domain toco:User ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:UserDevice ;
    :inverseOf toco:BelongsTo .

toco:hasDistance a :DatatypeProperty ;
    rdfs:comment "the Euclidean distance between the LiFi user equipment and access
        point."^^xsd:string ;
    rdfs:domain toco:LiFiAssociation ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAssociationProperty .

toco:hasErrorCode a :DatatypeProperty ;
    rdfs:comment "the error code of a service."^^xsd:string ;
    rdfs:domain toco:Service ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasFieldOfView a :DatatypeProperty ;
    rdfs:comment """Field of view (FOV) is the open observable area where a LiFi user
         equipment could receive visiable light signal. It is defined by let the
        incident angle of the LiFi user equipment equals to Pi/2.

If the incident angle larger than Pi/2, the light ray is out of the FOV, thus cannot
    be received by the LiFi user equipment."""^^xsd:string ;
    rdfs:domain toco:LiFiUserEquipment ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiUserEquipmentProperty .

toco:hasFlow a :ObjectProperty ;
```

```
    rdfs:domain toco:SystemDevice ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Flow .

toco:hasFlowAction a :ObjectProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Action .

toco:hasGainOfConcentrator a :DatatypeProperty ;
    rdfs:comment "The gain of the concentrator on the LiFi user equipment, usually
        equals to 1."^^xsd:string ;
    rdfs:domain toco:LiFiUserEquipment ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiUserEquipmentProperty .

toco:hasGainofOpticalFilter a :DatatypeProperty ;
    rdfs:comment "Gain of optical filter on LiFi access point."^^xsd:string ;
    rdfs:domain toco:LiFiAccessPoint ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAccessPointProperty .

toco:hasHalfIntensityAngle a :DatatypeProperty ;
    rdfs:comment """The angle of half intensity of the LED light transmiting visible
        light signal in a LiFi access point.

A fixed value. It is the angle where the radiated intensity will be half as intense
    as it would be at 0 degrees, or pointing straight ahead.

Factors that contribute to the angle of half intensity include the amount of
    diffusing material in the epoxy, the shape of the reflector cup which surrounds
    the LED chip, the shape of the LED lens, the distance from the LED to the tip of
    the lens, and the type of emitter chip.

The most common value is Pi/3."""^^xsd:string ;
    rdfs:domain toco:LiFiAccessPoint ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAccessPointProperty .

toco:hasIP a :DatatypeProperty ;
    rdfs:comment "the IP address of the interface."^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .
```

```
toco:hasIncidentAngle a :DatatypeProperty ;
    rdfs:comment "The angle of incidence of the LED receiver on LiFi user equipment,
        which is the angle between the visiable light ray incident on a surface and
        the line perpendicular to the surface at the point of incidence, called the
        normal."^^xsd:string ;
    rdfs:domain toco:LiFiAssociation ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAssociationProperty .

toco:hasInterfaceName a :DatatypeProperty ;
    rdfs:comment "the name of an interface"^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasJoinDate a :DatatypeProperty ;
    rdfs:comment "the join date of the user device" ;
    rdfs:domain toco:UserDevice ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:dateTime .

toco:hasLTEAssociation a :ObjectProperty ;
    rdfs:range toco:LTEAssociation ;
    rdfs:subPropertyOf toco:hasWirelessAssociation .

toco:hasLiFiAssociation a :ObjectProperty ;
    rdfs:domain toco:LiFiWLAN ;
    rdfs:range toco:LiFiAssociation ;
    rdfs:subPropertyOf toco:hasWirelessAssociation .

toco:hasLiFiWLAN a :ObjectProperty ;
    rdfs:domain toco:Device ;
    rdfs:range toco:LiFiWLAN ;
    rdfs:subPropertyOf toco:hasWLAN .

toco:hasMAC a :DatatypeProperty ;
    rdfs:comment "the MAC address of the interface."^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasOpticalTransmittedPower a :DatatypeProperty ;
    rdfs:comment "the transmitted power of the LED on the LiFi access point"^^xsd:
        string ;
    rdfs:domain toco:LiFiAccessPoint ;
```

```
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAccessPointProperty .

toco:hasPacketLoss a :DatatypeProperty ;
    rdfs:comment "the packet loss rate of a link, usually in percentage."^^xsd:string
        ;
    rdfs:domain toco:Link ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float .

toco:hasPort a :ObjectProperty ;
    rdfs:domain toco:Device ;
    rdfs:range toco:Interface ;
    rdfs:subPropertyOf toco:hasInterface .

toco:hasRadianceAngle a :DatatypeProperty ;
    rdfs:comment "the radiance angle of the light ray transmitted from LED on LiFi
        access point."^^xsd:string ;
    rdfs:domain toco:LiFiAssociation ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAssociationProperty .

toco:hasRespansivity a :DatatypeProperty ;
    rdfs:domain toco:LiFiAccessPoint ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLiFiAccessPointProperty .

toco:hasRoundTripTime a :ObjectProperty ;
    rdfs:comment "the round trip time of a link, which is the time taken for signal
        travel a round trip via this link."^^xsd:string ;
    rdfs:domain toco:Link ;
    rdfs:range om:ObervationAndMeasurement ;
    rdfs:subPropertyOf toco:hasLinkProperty .

toco:hasSatelliteAssociation a :ObjectProperty ;
    rdfs:domain toco:Device ;
    rdfs:range toco:SatelliteAssociation ;
    rdfs:subPropertyOf toco:hasWirelessAssociation .

toco:hasService a :ObjectProperty ;
    rdfs:domain toco:UserDevice ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Service .
```

```
toco:hasServiceEndTime a :DatatypeProperty ;
    rdfs:comment "the time that a service ends."^^xsd:string ;
    rdfs:domain toco:Service ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:dateTime .

toco:hasServiceStartTime a :DatatypeProperty ;
    rdfs:comment "the start time of a service."^^xsd:string ;
    rdfs:domain toco:Service ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:dateTime .

toco:hasServiceStatus a :DatatypeProperty ;
    rdfs:comment "the status of the service."^^xsd:string ;
    rdfs:domain toco:Service ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasServiceType a :DatatypeProperty ;
    rdfs:comment "the type of communication services, e.g., video, audio, file
        transmit, text transmiting, etc."^^xsd:string ;
    rdfs:domain toco:Service ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasStandardDeviation a :ObjectProperty ;
    rdfs:comment "After multiple round trip time tested on a link, the deviation of
        these test results."^^xsd:string ;
    rdfs:domain toco:Link ;
    rdfs:range om:ObervationAndMeasurement ;
    rdfs:subPropertyOf toco:hasLinkProperty .

toco:hasThroughPut a :ObjectProperty ;
    rdfs:domain toco:Link ;
    rdfs:range om:ObervationAndMeasurement ;
    rdfs:subPropertyOf toco:hasLinkProperty .

toco:hasTransmitPower a :ObjectProperty ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range om:ObervationAndMeasurement ;
    rdfs:subPropertyOf :topObjectProperty .

toco:hasUserDeviceStatus a :ObjectProperty ;
    rdfs:domain toco:UserDevice ;
    rdfs:range toco:UserDeviceStatus .
```

```
toco:hasUserID a :DatatypeProperty ;
    rdfs:comment "the user ID of the user" ;
    rdfs:domain toco:User ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:hasWiFiAssociation a :ObjectProperty ;
    rdfs:domain toco:WiFiWLAN ;
    rdfs:range toco:WiFiAssociation ;
    rdfs:subPropertyOf toco:hasWirelessAssociation .

toco:hasWiFiWLAN a :ObjectProperty ;
    rdfs:domain toco:Device ;
    rdfs:range toco:WiFiWLAN ;
    rdfs:subPropertyOf toco:hasWLAN .

toco:hasWiredLink a :ObjectProperty ;
    rdfs:domain toco:Interface ;
    rdfs:range toco:WiredLink ;
    rdfs:subPropertyOf toco:hasLink ;
    :propertyDisjointWith toco:hasWirelessAssociation .

toco:idle_timeout a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:isIn a :ObjectProperty ;
    rdfs:comment "the relation that a interface belongs to a device. A device can
        have multiple interfaces, but a interface can be in one and only one device
        ."^^xsd:string ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Device ;
    :inverseOf toco:hasInterface .

toco:isUP a :DatatypeProperty ;
    rdfs:comment "indicate whether a interface is up or not." ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:boolean .

toco:license a :AnnotationProperty .

toco:priority a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
```

```
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:range a :ObjectProperty ;
    rdfs:comment "the signal range of wireless devices, such as WiFi access point,
        LiFi access point, WiFi mobile station, and LiFi mobile station."^^xsd:string
         ;
    rdfs:domain toco:Device ;
    rdfs:isDefinedBy toco: ;
    rdfs:range om:ObervationAndMeasurement .

toco:ssid a :DatatypeProperty ;
    rdfs:comment "in wireless networks, the ssid of the network this device belongs
        to."^^xsd:string ;
    rdfs:domain toco:Device ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string .

toco:stationsInRange a :ObjectProperty ;
    rdfs:comment "in wireless networks (WiFi/LiFi), the mobile stations in the range
        of the access points."^^xsd:string ;
    rdfs:domain toco:AccessPoint ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:UserEquipment .

toco:table_id a :DatatypeProperty ;
    rdfs:domain toco:Flow ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:string ;
    rdfs:subPropertyOf toco:hasFlowProperty .

toco:toPort a :ObjectProperty ;
    rdfs:comment "the port to send to, defined in an flow action \"Output\" and \"
        Forward\"" ;
    rdfs:domain toco:Action ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Interface .

toco:unReachable a :Class ;
    rdfs:subClassOf toco:UserDeviceStatus .

<http://www.w3.org/2003/01/geo/wgs84_pos#Point> a :Class ;
    rdfs:label "location" ;
    rdfs:comment """The relation between something and the point,
 or other geometrical thing in space, where it is.  For example, the realtionship
     between
```

```
 a radio tower and a Point with a given lat and long.
 Or a relationship between a park and its outline as a closed arc of points, or a
     road and
 its location as a arc (a sequence of points).
 Clearly in practice there will be limit to the accuracy of any such statement, but
     one would expect
 an accuracy appropriate for the size of the object and uses such as mapping.""" ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#> ;
    rdfs:subClassOf <http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> .

<http://www.w3.org/2003/01/geo/wgs84_pos#alt> a :DatatypeProperty ;
    rdfs:comment "the altitude of a spartial thing, i.e., a point." ;
    rdfs:domain <http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#> ;
    rdfs:range xsd:float .

<http://www.w3.org/2003/01/geo/wgs84_pos#lat> a :DatatypeProperty ;
    rdfs:domain <http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#>,
        "the latitude of a point" ;
    rdfs:range xsd:float .

<http://www.w3.org/2003/01/geo/wgs84_pos#location> a :ObjectProperty ;
    rdfs:comment """The relation between something and the point,
 or other geometrical thing in space, where it is.  For example, the realtionship
     between
 a radio tower and a Point with a given lat and long.
 Or a relationship between a park and its outline as a closed arc of points, or a
     road and
 its location as a arc (a sequence of points).
 Clearly in practice there will be limit to the accuracy of any such statement, but
     one would expect
 an accuracy appropriate for the size of the object and uses such as mapping.""" ;
    rdfs:domain toco:Device ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#> ;
    rdfs:range <http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> .

<http://www.w3.org/2003/01/geo/wgs84_pos#long> a :DatatypeProperty ;
    rdfs:domain <http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#>,
        "the longitude of a spartial thing, i.e., a point." ;
    rdfs:range xsd:float .

foaf:firstName a :DatatypeProperty ;
    rdfs:comment "the first name of a user." ;
    rdfs:domain foaf:Person ;
    rdfs:isDefinedBy foaf: ;
```

```
    rdfs:range rdfs:Literal .

foaf:gender a :DatatypeProperty ;
    rdfs:comment "the gender of a user, reused from foaf" ;
    rdfs:domain foaf:Person ;
    rdfs:isDefinedBy foaf: ;
    rdfs:range rdfs:Literal .

foaf:lastName a :DatatypeProperty ;
    rdfs:comment """the last name of a person.
Reused from foaf""" ;
    rdfs:domain foaf:Person ;
    rdfs:isDefinedBy foaf: ;
    rdfs:range rdfs:Literal .

<http://purl.obolibrary.org/obo/UO_0000058> a :Class ;
    rdfs:label "rotational frequency unit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/UO_0000105> .

<http://purl.obolibrary.org/obo/UO_0000113> a :Class ;
    rdfs:label "power unit" ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

toco:BelongsTo a :ObjectProperty ;
    rdfs:domain toco:UserDevice ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:User .

toco:LTEAssociation a :Class ;
    rdfs:comment "the electromagnetic wave carring modulated signal between LTE base
        station, or eNodeB, and user equipment"^^xsd:string ;
    rdfs:subClassOf toco:WirelessAssociation .

toco:SatelliteAssociation a :Class ;
    rdfs:comment "the milli-meter electromagnetic wave carring modulated signal
        between satellite base stations and ground station."^^xsd:string ;
    rdfs:subClassOf toco:WirelessAssociation .

toco:WiFiAssociation a :Class ;
    rdfs:comment "the electromagnetic wave carring modulated signal between LTE base
        station, or eNodeB, and user equipment"^^xsd:string ;
    rdfs:subClassOf toco:WirelessAssociation .

toco:WiredLink a :Class ;
    rdfs:comment "wired link through twisted cable, optical fiber, etc." ;
    rdfs:subClassOf toco:Link .
```

```
toco:hasWirelessLinkProperty a :DatatypeProperty ;
    rdfs:domain toco:WirelessAssociation ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasLinkProperty .

<http://purl.obolibrary.org/obo/UO_0000111> a :Class ;
    rdfs:label "energy unit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

toco:LiFiWLAN a :Class ;
    rdfs:comment "WLAN interface on the devices in LIFi network, e.g., LiFi user
        equipment, LiFi access point."^^xsd:string ;
    rdfs:subClassOf toco:WLAN .

toco:Session a :Class ;
    rdfs:subClassOf toco:ConceptEntities .

toco:WiFiWLAN a :Class ;
    rdfs:comment "WLAN interface on the devices in LIFi network, e.g., LiFi user
        equipment, LiFi access point."^^xsd:string ;
    rdfs:subClassOf toco:WLAN .

toco:hasLiFiUserEquipmentProperty a :DatatypeProperty ;
    rdfs:comment "properties of LiFi user equipments."^^xsd:string ;
    rdfs:domain toco:LiFiUserEquipment ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float .

toco:hasLink a :ObjectProperty ;
    rdfs:domain toco:Interface ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Link .

toco:hasWLAN a :ObjectProperty ;
    rdfs:domain toco:Device ;
    rdfs:range toco:WLAN ;
    rdfs:subPropertyOf toco:hasInterface .

<http://purl.obolibrary.org/obo/UO_0000001> a :Class ;
    rdfs:label "length unit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

<http://purl.obolibrary.org/obo/UO_0000060> a :Class ;
    rdfs:label "speed/velocity unit" ;
```

```
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

<http://purl.obolibrary.org/obo/UO_0000105> a :Class ;
    rdfs:label "frequency unit" ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

toco:LiFiUserEquipment a :Class ;
    rdfs:comment "any device used directly by an end-user to communicate. It can be a
         hand-held telephone, a laptop computer equipped with a mobile broadband
        adapter, or any other device."^^xsd:string ;
    rdfs:subClassOf toco:UserEquipment .

toco:User a :Class ;
    rdfs:comment "reused from foaf (http://xmlns.com/foaf/0.1/). Represent a user in
        the telecommunication system."^^xsd:string ;
    rdfs:subClassOf foaf:Person .

toco:hasInterface a :ObjectProperty ;
    rdfs:comment "the fact that a device has a interface."^^xsd:string ;
    rdfs:domain toco:Device ;
    rdfs:isDefinedBy toco: ;
    rdfs:range toco:Interface .

toco:hasLiFiAssociationProperty a :DatatypeProperty ;
    rdfs:comment "The properties of the visiable light association between the LED on
         LiFi access point and LiFi user equipment."^^xsd:string ;
    rdfs:domain toco:LiFiAssociation ;
    rdfs:range xsd:float ;
    rdfs:subPropertyOf toco:hasWirelessLinkProperty .

toco:PhysicalInfrastructure a :Class ;
    rdfs:comment """Describe all the physical resources in a telecommunication
        network system, which is divided into three main modules, namely, device,
        interface, link.

Please be reminded the electromagnetic wave (which is the links of all wireless
    communication netwoks), even if it is invisable, physically exists. Thus, it is
    also described in the subclass of "net;PhysicalInfrastructure"."""^^xsd:string .

toco:UserDeviceStatus a :Class ;
    rdfs:subClassOf toco:ConceptEntities .

toco:hasLiFiAccessPointProperty a :DatatypeProperty ;
    rdfs:comment "datatype properties dedicated for LiFi access points"^^xsd:string ;
    rdfs:domain toco:LiFiAccessPoint ;
    rdfs:isDefinedBy toco: ;
```

```
    rdfs:range xsd:float .

foaf:Person a :Class ;
    rdfs:isDefinedBy foaf: ;
    rdfs:subClassOf toco:ConceptEntities .

<http://purl.obolibrary.org/obo/UO_0000231> a :Class ;
    rdfs:label "information unit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

toco:Action a :Class ;
    rdfs:comment "the action of a flow defined in OpenFlow protocol" ;
    rdfs:subClassOf toco:ConceptEntities .

toco:LiFiAccessPoint a :Class ;
    rdfs:comment "The access point in Li-Fi network. Li-Fi is a wireless optical
        networking technology that uses light-emitting diodes (LEDs) for data
        transmission."^^xsd:string ;
    rdfs:subClassOf toco:AccessPoint .

toco:LiFiAssociation a :Class ;
    rdfs:comment "the visiable light carring modulated signals in Li-Fi network.
        Transmited and received by LiFi user equipment and LiFi access point."^^xsd:
        string ;
    rdfs:subClassOf toco:WirelessAssociation .

toco:SystemDevice a :Class ;
    rdfs:comment "All the physical facilities in the telecommunication system that
        provide the communication service to uses. For example: base station towers,
        Wi-Fi access points, switches, servers, etc."^^xsd:string ;
    rdfs:subClassOf toco:Device .

toco:UserEquipment a :Class ;
    rdfs:comment "any device used directly by an end-user to communicate. It can be a
         hand-held telephone, a laptop computer equipped with a mobile broadband
        adapter, or any other device."^^xsd:string ;
    rdfs:subClassOf toco:UserDevice .

toco:hasLinkProperty a :DatatypeProperty,
        :ObjectProperty ;
    rdfs:domain toco:Link ;
    rdfs:isDefinedBy toco: ;
    rdfs:range xsd:float .

toco:hasWirelessAssociation a :ObjectProperty ;
    rdfs:domain toco:WLAN ;
```

```
    rdfs:range toco:WirelessAssociation ;
    rdfs:subPropertyOf toco:hasLink .

<http://www.w3.org/2003/01/geo/wgs84_pos#SpartialThing> a :Class ;
    rdfs:isDefinedBy <http://www.w3.org/2003/01/geo/wgs84_pos#> ;
    rdfs:subClassOf toco:ConceptEntities .

toco:AccessPoint a :Class ;
    rdfs:comment "a wireless access point, or more generally just access point (AP),
        is a networking hardware device that allows a Wi-Fi/Li-Fi device to connect
        to a wired network. The AP usually connects to a router (via a wired network)
         as a standalone device, but it can also be an integral component of the
        router itself. An AP is differentiated from a hotspot, which is the physical
        location where Wi-Fi access to a WLAN is available."^^xsd:string ;
    rdfs:subClassOf toco:SystemDevice .

toco:WLAN a :Class ;
    rdfs:comment "A wireless local area network (WLAN) interface, also known as WLAN,
         is the interface of a wireless computer network that links two or more
        devices using wireless communication within a limited area such as a home,
        school, computer laboratory, or office building." ;
    rdfs:subClassOf toco:Interface .

toco:WirelessAssociation a :Class ;
    rdfs:comment "the electromeganetic radio transmiting through wireless channels in
         wireless network, carring communication data. Depend on its frequency, it
        could be visiable light, micro waves, radio waves."^^xsd:string ;
    rdfs:subClassOf toco:Link .

toco:hasFlowProperty a :DatatypeProperty ;
    rdfs:isDefinedBy toco: .

<http://purl.obolibrary.org/obo/UO_0000003> a :Class ;
    rdfs:label "time unit" ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf <http://purl.obolibrary.org/obo/uo.owl/unit> .

<http://purl.obolibrary.org/obo/uo.owl/unit> a :Class ;
    rdfs:comment "reused from Units of measurement Ontology"^^xsd:string ;
    rdfs:isDefinedBy <http://purl.obolibrary.org/obo/uo.owl> ;
    rdfs:subClassOf toco:ConceptEntities .

om:ObervationAndMeasurement a :Class ;
    rdfs:comment "reused from SenSei Observation And Measurement Ontology (http://
        purl.oclc.org/net/unis/ontology/sensordata.owl)."^^xsd:string ;
    rdfs:isDefinedBy <http://purl.oclc.org/net/unis/ontology/sensordata.owl> ;
    rdfs:subClassOf toco:ConceptEntities .
```

```
toco:ConceptEntities a :Class ;
    :disjointWith toco:PhysicalInfrastructure .

toco:Flow a :Class ;
    rdfs:comment "the flow entry in a flow table inside a switch or route. Defined by
         OpenFlow protocol." ;
    rdfs:subClassOf toco:ConceptEntities .

toco:UserDevice a :Class ;
    rdfs:comment "User Devices, requiring connectivity services, e.g., phones,
        tablets, laptops, wearable devices, etc."^^xsd:string ;
    rdfs:subClassOf toco:Device .

toco:Service a :Class ;
    rdfs:comment "a communication service, e.g., voice, video, file uploading/
        downloading, etc."^^xsd:string ;
    rdfs:subClassOf toco:ConceptEntities .

toco:Link a :Class ;
    rdfs:subClassOf toco:PhysicalInfrastructure .

toco:Device a :Class ;
    rdfs:comment """represent all the devices in the physical infrastructure of
        telecommunciation system.

According to its user, could be divided into UserDevice and SystemDevice."""^^xsd:
    string ;
    rdfs:subClassOf toco:PhysicalInfrastructure .

toco:Interface a :Class ;
    rdfs:subClassOf toco:PhysicalInfrastructure .

toco: a :Ontology ;
    rdfs:comment """Toucan Ontology (ToCo)

A ontology developed for telecommunciation network systems with hybrid technologies,
    e.g., WiFi, LiFi, LTE, 2G/3G, optical, etc.""" ;
    rdfs:isDefinedBy "Qianru Zhou (qz1@hw.ac.uk)"^^xsd:string ;
    :versionIRI toco: ;
    toco2:contributor "Mohammad Soltani Dehghani (for LiFi networks)"^^xsd:string ;
    toco2:license "https://www.w3.org/Consortium/Legal/2015/copyright-software-and-
        document"^^xsd:anyURI .
```

Appendix C

The SPQRAL Query Files

C.1 The SPARQL Query for Inconsistent Flows

```
PREFIX ns: <http://purl.org/toco/>
select distinct ?in_port ?to_port
where {
?s a ns:Switch; ns:hasFlow ?f.
?f a ns:PathFlow;
        ns:in_port ?in_p;
        ns:hasAction ?a.
?a ns:toPort ?to_p.

bind ( strafter(str(?s), "http://purl.org/toco/") as ?switch).
bind ( concat(str(?switch) , '-eth', str(?in_p)) as ?in_port ).
bind ( concat(str(?switch) , '-eth', str(?to_p)) as ?to_port ).

?p1 a ns:Port;
   ns:hasName ?in_port;
   ns:isUP ?isUp1.

?p2 a ns:Port;
   ns:hasName ?to_port;
   ns:isUP ?isUp2.

filter (?isUp1 = "false"^^xsd:boolean || ?isUp2 = "false"^^xsd:boolean)
}
```

C.2 The SPARQL Query for Lost Silence

```
REGISTER QUERY StreamingAndExternalStaticRdfGraph AS
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pos: <http://www.w3.org/2003/01/geo/wgs84_pos/>
PREFIX net: <http://purl.org/toco/>
PREFIX fn:<http://www.w3.org/2005/xpath-functions#>
SELECT (COUNT(?UE) AS ?counter) ?lat ?long
FROM STREAM <http://purl.org/toco/stream1> [RANGE 30m STEP 5s]
FROM <http://www.w3.org/2005/xpath-functions>
WHERE {
    ?UE net:hasStatus net:unReachable.
    ?UE pos:location ?point.
    ?point pos:lat ?lat;
    pos:long ?long.
    BIND (fn:round(?lat * 1000) as ?roundLat)
    BIND (fn:round(?long * 1000) as ?roundLong)
}
GROUP BY (ROUND(?lat) AS ?roundlat) (ROUND(?long) AS ?roundlong)
GROUP BY ?lat ?long HAVING (?counter >10)
```

C.3 The SPARQL Query for SARA

```
PREFIX : <http://purl.org/toco/>
SELECT  ?ap  (COUNT(?asso) AS ?cnt)
WHERE {
    ?aps :stationsInRange :sta1;
        :associatedStations ?asso; :hasWLAN ?w.
    ?w :antennaGain ?g; :hasTxPower ?tx.
    ?asso a :Association; :From ?aps; :To :sta1;
         :hasBandWidth ?bw .
    ?bw :hasValue ?bwValue .
    bind(strafter(str(?aps), "http://purl.org/toco/") as ?ap) .
}
GROUP BY ?aps
ORDER BY ?cnt DESC(?bwValue) DESC(?g) DESC(?tx)
```

References

[1] Marshall T Rose. Management information base for network management of tcp/ip-based internets: Mib-ii. *Management*, 1990.

[2] Rob Sherwood, Glen Gibb, and Masayoshi Kobayashi. Carving research slices out of your production networks with openflow. *ACM SIGCOMM Computer Communication Review*, 40(1):129–130, 2010.

[3] Cheng-Xiang Wang, Fourat Haider, Xiqi Gao, Xiao-Hu You, Yang Yang, Dongfeng Yuan, Hadi Aggoune, Harald Haas, Simon Fletcher, and Erol Hepsaydir. Cellular architecture and key technologies for 5g wireless communication networks. *IEEE Communications Magazine*, 52(2):122–130, 2014.

[4] Scott Shenker, Martin Casado, Teemu Koponen, and Nick McKeown. The future of networking, and the past of protocols. *Open Networking Summit*, 20:1–30, 2011.

[5] Jennifer Rexford. The networking philosopher's problem. *ACM SIGCOMM Computer Communication Review*, 41(3):5–9, 2011.

[6] Zied Ben Houidi. A knowledge-based systems approach to reason about networking. In *ACM Workshop on Hot Topics in Networks*, pages 22–28, 2016.

[7] David Benyon. *Information and data modelling*. McGraw-Hill Higher Education, 1996.

[8] Freek Dijkstra, Bert Andree, Karst Koymans, and Jeroen van der Ham. Introduction to itu-t recommendation g. 805, 2007.

[9] David Cleary, Boris Danev, and Diarmuid O'Donoghue. Using ontologies to simplify wireless network configuration. In *FOMI*, 2005.

[10] Claudia Villalonga, Martin Strohbach, Niels Snoeck, Michael Sutterer, Mariano Belaunde, Ernö Kovacs, Anna V Zhdanova, Laurent Walter Goix, and Olaf Droegehorn. Mobile ontology: Towards a standardized semantic model for the mobile domain. In *International Conference on Service-Oriented Computing*, pages 248–257. Springer, 2007.

[11] Ann Devitt, Boris Danev, and Katarina Matusikova. Ontology-driven automatic construction of bayesian networks for telecommunication network management. In *FOMI*, 2006.

[12] Jorge E López De Vergara, Antonio Guerrero, Víctor A Villagrá, and Julio Berrocal. Ontology-based network management: study cases and lessons learned. *Journal of Network and Systems Management*, 17(3):234–254, 2009.

[13] Antonio Guerrero, Víctor A Villagrá, Jorge E López De Vergara, and Julio Berrocal. Ontology-based integration of management behaviour and information definitions using swrl and owl. In *DSOM*, pages 12–23. Springer, 2005.

[14] John Strassner, José Neuman De Souza, Sven Van der Meer, Steven Davy, Keara Barrett, David Raymer, and Srini Samudrala. The design of a new policy model to support ontology-driven reasoning for autonomic networking. *Journal of Network and Systems Management*, 17(1-2):5–32, 2009.

[15] Debao Xiao and Hui Xu. An integration of ontology-based and policy-based network management for automation. In *IEEE International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pages 27–27, 2006.

[16] Jeroen Johannes van der Ham. *A semantic model for complex computer networks: the network description language*, volume 3. Citeseer, 2010.

[17] A Bierman, M Bjorklund, J Dong, and D Romascanu. A yang data model for hardware management. Technical report, 2018.

[18] Martin Bjorklund. A yang data model for ip management. 2018.

[19] Martin Bjorklund. A yang data model for snmp configuration. 2014.

[20] Liam Fallon and Declan O'Sullivan. Using a semantic knowledge base for communication service quality management in home area networks. In *IEEE Network Operations and Management Symposium*, pages 43–51, 2012.

[21] Ryota Ohshima, Shinji Kawaguchi, Osamu Kamatani, Osamu Akashi, Kunitake Kaneko, and Fumio Teraoka. Construction of routing information knowledge base towards wide area network management. In *ACM the 10th International Conference on Future Internet*, pages 76–83, 2015.

[22] David D Clark, Craig Partridge, J Christopher Ramming, and John T Wroclawski. A knowledge plane for the internet. In *ACM SIGCOMM*, pages 3–10, 2003.

[23] Asunción Gómez-Pérez and Richard Benjamins. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. IJCAI CEUR Workshop, 1999.

[24] Reid G Smith. Knowledge-based systems: Concepts, techniques, examples. *Canadian High Technology Show*, 1985.

[25] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[26] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[27] Dieter Fensel. Ontology-based knowledge management. *Computer*, 35(11):56–59, 2002.

[28] Thomas R Gruber. The role of common ontology in achieving sharable, reusable knowledge bases. pages 601–602. KR&R, Morgan Kaufrnann, 1991.

[29] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International journal of human-computer studies*, 43(5-6):625–640, 1995.

[30] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[31] Igor Jurisica, John Mylopoulos, and Eric Yu. Ontologies for knowledge management: an information systems perspective. *Knowledge and Information systems*, 6(4):380–401, 2004.

[32] Haruhiko Kaiya and Motoshi Saeki. Using domain ontology as domain knowledge for requirements elicitation. In *IEEE Requirements Engineering*, pages 189–198, 2006.

[33] Suyeon Kim, Euiho Suh, and Hyunseok Hwang. Building the knowledge map: an industrial case study. *Journal of knowledge management*, 7(2):34–45, 2003.

[34] Liana Razmerita, Albert Angehrn, and Alexander Maedche. Ontology-based user modeling for knowledge management systems. *User modeling*, pages 148–148, 2003.

[35] York Sure and Rudi Studer. A methodology for ontology-based knowledge management. *Towards the semantic web: Ontology-driven knowledge management*, pages 33–46, 2003.

[36] Frederick Hayes-Roth, Donald Waterman, and Douglas Lenat. Building expert systems. 1984.

[37] Grigoris Antoniou and Frank Van Harmelen. *A semantic web primer*. MIT press, 2004.

[38] Qianru Zhou, Alasdair J. G. Gray, and Stephen McLaughlin. Toco: An ontology for hybrid telecommunication networks. *Journal of Semantic Web*, submitted.

[39] Qianru Zhou, Stephen McLaughlin, Alasdair J. G. Gray, Shangbin Wu, and Chengxiang Wang. Lost silence: An emergency response early detection service through continuous processing of telecommunication data streams. In *ISWC Web Stream Processing Workshop 2017*, pages 33–47.

[40] Michael Behringer, Steinthor Bjarnason, Sheng Jiang, Brian Carpenter, Max Pritikin, Laurent Ciavaglia, and Alex Clemm. Autonomic networking: Definitions and design goals. 2015.

[41] Uyless D Black. *Network Management Standards: SNMP, CMIP, TMN, MIBs and Object Libraries*. McGraw-Hill, Inc., 1994.

[42] Jeffrey D Case, Mark Fedor, Martin L Schoffstall, and James Davin. Simple network management protocol (snmp). Technical report, 1990.

[43] Why use NETCONF/YANG when you can use SNMP and CLI. Available: https://www.snmpcenter.com/why-use-netconf/.

[44] David Harrington, Bert Wijnen, and Randy Presuhn. An architecture for describing simple network management protocol (snmp) management frameworks. 2002.

[45] David Perkins and Evan McGinnis. *Understanding SNMP MIBs*. Prentice Hall PTR Englewood Cliffs, 1997.

[46] Randy Presuhn. Management information base (mib) for the simple network management protocol (snmp). *RFC 3418*, 2002.

[47] William Stallings. Snmp and snmpv2: the infrastructure for network management. *IEEE Communications Magazine*, 36(3):37–43, 1998.

[48] William Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., 1998.

[49] J Schoenwaelder. Overview of the 2002 iab network management workshop. *RFC 3535*, May, 2003.

[50] Mark A Bach, Kyle Jeffrey Charlet, Shyh-Mei Fang Ho, Kevin M McBride, Huey Moncrief Rowe-Anderson, Thomas Beavers Sander, and Thomas Arthur Vogel. Command line interface for creating business objects for accessing a hierarchical database, Oct. 2000. US Patent 6,141,660.

[51] Shehzad T Merchant, Manish M Rathi, Victor C Lin, Vipin K Jain, Jia-Ru Li, Amit K Maitra, Matthew R Peters, Derek H Pitcher, and Balaji Srinivasan. Unified, configurable, adaptive, network architecture, Dec. 2007. US Patent 7,310,664.

[52] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, 2013.

[53] Martin Bjorklund. A yang data model for interface management. *RFC 7223*, 2014.

[54] Martin Bjorklund. A yang data model for ip management. *RFC 8344*, 2014.

[55] Phil Shafer. An architecture for network management using netconf and yang. *RFC 6244*, 2011.

[56] Prasad Calyam, Lakshmi Kumarasamy, Chang-Gun Lee, and Fusun Ozguner. Ontology-based semantic priority scheduling for multi-domain active measurements. *Journal of Network and Systems Management*, 22(3):331–365, 2014.

[57] Alfonso Castro, Víctor A Villagrá, Beatriz Fuentes, and Begona Costales. A flexible architecture for service management in the cloud. *IEEE transactions on Network and Service Management*, 11(1):116–125, 2014.

[58] Talita De Paula Cypriano De Souza, Christian Esteve Rothenberg, Mateus Augusto Silva Santos, and Luciano Bernardes De Paula. Towards semantic network models via graph databases for sdn applications. In *IEEE EWSDN*, pages 49–54, 2015.

[59] Liam Fallon and Declan O'Sullivan. Aesop: A semantic system for autonomic management of end-user service quality. In *IFIP/IEEE IM 2013*, pages 716–719.

[60] Fangwen Fu and Ulaº C Kozat. Wireless network virtualization as a sequential auction game. In *IEEE INFOCOM 2010*, pages 1–9.

[61] KL Eddie Law and Achint Saxena. Upm: Unified policy-based network management. In *ITCom SPIE 2001*, volume 4523, pages 326–337.

[62] Nancy Samaan and Ahmed Karmouch. Towards autonomic network management: an analysis of current and future research directions. *IEEE Communications Surveys & Tutorials*, 11(3), 2009.

[63] Jeroen J Van Der Ham, Paola Grosso, Freek Dijkstra, Arie Taal, and CTAM de Laat. On the impact of network topology aggregation in multi-domain lightpath provisioning. *System and Network Engineering Technical Report SNE-UVA-2010-2*, 2010.

[64] Dinesh C Verma. Simplifying network administration using policy-based management. *IEEE network*, 16(2):20–26, 2002.

[65] Hui Xu and Debao Xiao. Applying semantic web services to automate network management. In *IEEE ICIEA*, pages 461–466, 2007.

[66] Li Xu, Freek Dijkstra, Damien Marchal, Arie Taal, Paola Grosso, and Cees De Laat. A declarative approach to multi-layer path finding based on semantic network descriptions. In *IEEE ONDM 2009*, pages 1–6.

[67] Qianru Zhou, Cheng-Xiang Wang, Stephen McLaughlin, and Xiaotian Zhou. Network virtualization and resource description in software-defined wireless networks. *IEEE Communications Magazine*, 53(11):110–117, 2015.

[68] Hao Zhu, Xiangke Liao, Cees de Laat, and Paola Grosso. Joint flow routing-scheduling for energy efficient software defined data center networks: A prototype of energy-aware network management platform. *Journal of Network and Computer Applications*, 63:110–124, 2016.

[69] Edward A Felgenbaum. The art of artificial intelligence. *IJCAI*, 1977.

[70] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River, 2003.

[71] RS Aylett and Christophe Doniat. Supporting the domain expert in planning domain construction. In *the AIPS Workshop on Knowledge Engineering Tools and Techniques for AI Planning*, 2002.

[72] Richard Benjamins, Dieter Fensel, and Asunción Gómez-Pérez. Knowledge management through ontologies. CEUR-WS, 1998.

[73] Xu Chen, Yun Mao, Zhuoqing Morley Mao, and Jacobus E van der Merwe. Knowops: Towards an embedded knowledge base for network management and operations. In *Hot-ICE*, 2011.

[74] Peter Clark, John Thompson, and Bruce Porter. Knowledge patterns. In *Handbook on Ontologies*, pages 191–207. Springer, 2004.

[75] Asuncion Gomez-Perez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer Science & Business Media, 2006.

[76] Barry Smith. Mereotopology: a theory of parts and boundaries. *Data & Knowledge Engineering*, 20(3):287–303, 1996.

[77] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. *Ontology learning from text: methods, evaluation and applications*, volume 123. IOS press, 2005.

[78] Christiaan Fluit, Marta Sabou, and Frank Van Harmelen. Ontology-based information visualization: toward semantic web applications. In *Visualizing the semantic web*, pages 45–58. Springer, 2006.

[79] Clyde W Holsapple and Kshiti D Joshi. A formal knowledge management ontology: Conduct, activities, resources, and influences. *Journal of the Association for Information Science and Technology*, 55(7):593–612, 2004.

[80] Natalya F Noy and Deborah L McGuinness. Ontology development 101: A guide to creating your first ontology. *Stanford knowledge systems laboratory*, 2001.

[81] Natalya F Noy, Monica Crubézy, Ray W Fergerson, Holger Knublauch, Samson W Tu, Jennifer Vendetti, and Mark A Musen. Protege-2000: an open-source ontology-development and knowledge-acquisition environment. In *AMIA Annu Symp Proc*, volume 953, page 953, 2003.

[82] An He, Kyung Kyoon Bae, Timothy R Newman, Joseph Gaeddert, Kyouwoong Kim, Rekha Menon, Lizdabel Morales-Tirado, Youping Zhao, Jeffrey H Reed, and William H Tranter. A survey of artificial intelligence for cognitive radios. *IEEE Transactions on Vehicular Technology*, 59(4):1578–1592, 2010.

[83] Jim Hendler and Tim Berners-Lee. From the semantic web to social machines: A research challenge for ai on the world wide web. *Artificial Intelligence*, 174(2):156–161, 2010.

[84] Mariano Fernández-López and Asunción Gómez-Pérez. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17(2):129–156, 2002.

[85] Janez Brank, Marko Grobelnik, and Dunja Mladenić. A survey of ontology evaluation techniques. 2005.

[86] Asunción Gómez-Pérez. Evaluation of ontologies. *International Journal of intelligent systems*, 16(3):391–409, 2001.

[87] Maria Papadopouli, Elias Raftopoulos, and Haipeng Shen. Evaluation of short-term traffic forecasting algorithms in wireless networks. In *IEEE NGI*, pages 8–pp, 2006.

[88] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[89] Tim Berners-Lee. Semantic web road map, 1998.

[90] Tim Berners-Lee. What the semantic web can represent. 1998.

[91] Tim Berners-Lee, Mark Fischetti, and Michael L Foreword By-Dertouzos. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. HarperInformation, 2000.

[92] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[93] Tim Berners-Lee, Wendy Hall, James A Hendler, Kieron O'Hara, Nigel Shadbolt, and Daniel J Weitzner. A framework for web science. *Foundations and trends in Web Science*, 1(1):1–130, 2006.

[94] Tim Berners-Lee and Lalana Kagal. The fractal nature of the semantic web. *AI magazine*, 29(3):29, 2008.

[95] Tim Berners-Lee. Long live the web. *Scientific American*, 303(6):80–85, 2010.

[96] Sean Bechhofer. Owl: Web ontology language. In *Encyclopedia of Database Systems*, pages 2008–2009. Springer, 2009.

[97] Ilkka Niskanen and Julia Kantorovitch. Ontology driven data mining and information visualization for the networked home. In *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*, pages 147–156. IEEE, 2010.

[98] Sehl Mellouli, Faouzi Bouslama, and Aichath Akande. An ontology for representing financial headline news. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2):203–208, 2010.

[99] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web. In *ACM World Wide Web*, pages 1265–1266, 2008.

[100] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: Principles and state of the art. In *World wide web conference*, pages 1–40, 2008.

[101] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227, 2009.

[102] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *Intelligent Systems*, 21(3):96–101, 2006.

[103] Dieter Fensel, Holger Lausen, Axel Polleres, Jos De Bruijn, Michael Stollberg, Dumitru Roman, and John Domingue. *Enabling semantic web services: the web service modeling ontology*. Springer Science & Business Media, 2006.

[104] Ian Niles and Adam Pease. Towards a standard upper ontology. In *ACM FOIS*, pages 2–9, 2001.

[105] Pedro PF Barcelos, Maxwell E Monteiro, Ricardo de M Simões, Anilton S Garcia, and Marcelo EV Segatto. Ootn-an ontology proposal for optical transport networks. In *IEEE ICUMT*, pages 1–7, 2009.

[106] Abdulbaki Uzun and Axel Küpper. Openmobilenetwork: extending the web of data by a dataset for mobile networks and devices. In *ACM ICSS*, pages 17–24, 2012.

[107] Dean Jones, Trevor Bench-Capon, and Pepijn Visser. Methodologies for ontology development. In *IT and KNOWS Conference, XV IFIP World Computer Congress, Budapest*, Auguest, 1998.

[108] Camila Bezerra, Fred Freitas, and Filipe Santana. Evaluating ontologies with competency questions. In *IEEE/WIC/ACM WI IAT*, volume 3, pages 284–285, 2013.

[109] Giancarlo Guizzardi and Gerd Wagner. Using the unified foundational ontology (ufo) as a foundation for general conceptual modeling languages. In *Theory and Applications of Ontology: Computer Applications*, pages 175–196. Springer, 2010.

[110] Antonio Guerrero, Víctor A Villagrá, Jorge E López de Vergara, Alfonso Sánchez-Macián, and Julio Berrocal. Ontology-based policy refinement using

swrl rules for management information definitions in owl. *Lecture Notes in Computer Science*, 4269:227, 2006.

[111] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 258–267. IEEE, 2004.

[112] Hao Zhu, Karel van der Veldt, Zhiming Zhao, Paola Grosso, Dimitar Pavlov, Joris Soeurt, Xiangke Liao, and Cees de Laat. A semantic enhanced power budget calculator for distributed computing using ieee 802.3 az. *Cluster Computing*, 18(1):61–77, 2015.

[113] Jeff Z Pan. Resource description framework. In *Handbook on ontologies*, pages 71–90. Springer, 2009.

[114] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[115] Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1999.

[116] Eric Miller. An introduction to the resource description framework. *Bulletin of the Association for Information Science and Technology*, 25(1):15–19, 1998.

[117] Eric Prud and Andy Seaborne. Sparql query language for rdf. 2006.

[118] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM TODS*, 34(3):16, 2009.

[119] Víctor A Villagrá and Jorge E López De Vergara. Ontology based policy refinement using swrl rules for management information definitions. In *IEEE DSOM*, 2006.

[120] Bob Lantz, Brandon Heller, and Nick McKeown. A network in a laptop: rapid prototyping for software-defined networks. In *ACM SIGCOMM Workshop on Hot Topics in Networks*, page 19. ACM, 2010.

[121] Fei Hu, Qi Hao, and Ke Bao. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4):2181–2206, 2014.

[122] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[123] Rob Sherwood, Glen Gibb, Kok-Kiong Yap, Guido Appenzeller, Martin Casado, Nick McKeown, and Guru Parulkar. Flowvisor: A network virtualization layer. *OpenFlow Switch Consortium, Tech. Rep*, 1:132, 2009.

[124] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. Openroads: Empowering research in mobile networks. *ACM SIGCOMM Computer Communication Review*, 40(1):125–126, 2010.

[125] Dmitry A Drutskoy. *Software-defined network virtualization with flown.* PhD thesis, Princeton University, 2012.

[126] Marcelo R Nascimento, Christian E Rothenberg, Marcos R Salvador, Carlos NA Corrêa, Sidney C de Lucena, and Maurício F Magalhães. Virtual routers as a service: the routeflow approach leveraging software-defined networks. In *ACM CFI*, pages 34–37. ACM, 2011.

[127] Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. Softran: Software defined radio access network. In *ACM SIGCOMM HotSDN*, pages 25–30, 2013.

[128] Mao Yang, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. Openran: a software-defined ran architecture via virtualization. In *ACM SIGCOMM computer communication review*, volume 43, pages 549–550, 2013.

[129] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. Towards programmable enterprise wlans with odin. In *ACM HotSDN*, pages 115–120. ACM, 2012.

[130] Ramon Dos Reis Fontes, Claudia Campolo, Christian Esteve Rothenberg, and Antonella Molinaro. From theory to experimental evaluation: Resource management in software-defined vehicular networks. *IEEE Access*, 5:3069–3076, 2017.

[131] Naga Praveen Katta, Jennifer Rexford, and David Walker. Logic programming for software-defined networks. In *XLDI*, volume 412, 2012.

[132] Nguyen B Truong, Gyu Myoung Lee, and Yacine Ghamri-Doudane. Software defined networking-based vehicular adhoc network with fog computing. In *IFIP/IEEE IM*, pages 1202–1207. IEEE, 2015.

[133] Ramya Raghavendra, Jorge Lobo, and Kang-Won Lee. Dynamic graph query primitives for sdn-based cloudnetwork management. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 97–102. ACM, 2012.

[134] Annie Ibrahim Rana, Brendan Jennings, Mícheál Ó Foghlú, and Sven van der Meer. Autonomic policy-based han traffic classification using augmented meta model for policy translation. In *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*, pages 1–8. IEEE, 2011.

[135] Annie Ibrahim Rana and Brendan Jennings. Semantic uplift of monitoring data to select policies to manage home area networks. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 368–375. IEEE, 2012.

[136] Annie Ibrahim Rana and Brendan Jennings. Semantic aware processing of user defined inference rules to manage home networks. *Journal of Network and Computer Applications*, 79:68–87, 2017.

[137] John Strassner, Nazim Agoulmine, and Elyes Lehtihet. Focale: A novel autonomic networking architecture. In *LAACS, Campo Grande, MS, Brazil*, 2006.

[138] Jorge E López de Vergara, Víctor A Villagrá, Carlos Fadón, Juan M González, José A Lozano, and Manuel Álvarez-Campana. An autonomic approach to offer

services in osgi-based home gateways. *Computer Communications*, 31(13):3049–3058, 2008.

[139] JE Lopez De Vergara, Víctor A Villagrá, Julio Berrocal, Juan I Asensio, and Roney Pignaton. Semantic management: application of ontologies for the integration of management information models. In *IFIP/IEEE INM*, pages 131–134, 2003.

[140] José A Lozano, Alfonso Castro, Juan M González, Jorge E López De Vergara, Víctor A Villagrá, and Vicente Olmedo. Autonomic provisioning model for digital home services. In *IEEE MACE*, pages 114–119. Springer, 2008.

[141] Jorge E López de Vergara, Enrique Vázquez, and Javier Guerra. Security policy instantiation to react to network attacks-an ontology-based approach using owl and swrl. In *SECRYPT*, pages 78–83, 2008.

[142] Jorge E López de Vergara, Javier Aracil, Jesús Martínez, Alfredo Salvador, and José Alberto Hernández. Application of ontologies for the integration of network monitoring platforms. 2008.

[143] Dimitris Kanelopoulos. Ontology-based network management for autonomic communications. *Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology: Creating Information System Ecology*, page 9, 2009.

[144] Andrea Westerinen, John Schnizlein, John Strassner, Mark Scherling, Bob Quinn, Shai Herzog, A Huynh, Mark Carlson, Jay Perry, and Steve Waldbusser. Terminology for policy-based management. Technical report, 2001.

[145] Bob Moore, Ed Ellesson, John Strassner, and Andrea Westerinen. Policy core information model–version 1 specification. Technical report, 2001.

[146] Alfred Ka Yiu Wong, Pradeep Ray, Nandan Parameswaran, and John Strassner. Ontology mapping for the interoperability problem in network management. *IEEE Journal on selected areas in Communications*, 23(10):2058–2068, 2005.

[147] JF Overbeek. Meta object facility (mof): investigation of the state of the art. Master's thesis, University of Twente, 2006.

[148] Artur Boronat and José Meseguer. An algebraic semantics for mof. In *International Conference on Fundamental Approaches to Software Engineering*, pages 377–391. Springer, 2008.

[149] Uwe Aßmann, Steffen Zschaler, and Gerd Wagner. Ontologies, meta-models, and the model-driven paradigm. In *Ontologies for software engineering and software technology*, pages 249–273. Springer, 2006.

[150] Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1998.

[151] Mattijs Ghijsen, Jeroen Van Der Ham, Paola Grosso, Cosmin Dumitru, Hao Zhu, Zhiming Zhao, and Cees De Laat. A semantic-web approach for modeling computing infrastructures. *Computers & Electrical Engineering*, 39(8):2553–2565, 2013.

[152] Jeremy J Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *ACM World Wide Web*, pages 74–83, 2004.

[153] RDFLib Team. Python library rdflib, 2013.

[154] Stardog: the Enterprise Knowledge Graph. Available: http://www.stardog.com.

[155] Mininet Team. Mininet: An instant virtual network on your laptop (or other pc), 2012.

[156] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan J Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, and Amidon K. The design and implementation of open vswitch. In *NSDI*, pages 117–130, 2015.

[157] John Strassner. *Policy-based network management: solutions for the next generation.* Morgan Kaufmann, 2003.

[158] John Strassner, Sven van der Meer, Declan O'Sullivan, and Simon Dobson. The use of context-aware policies and ontologies to facilitate business-aware network management. *Journal of Network and Systems Management*, 17(3):255–284, 2009.

[159] R de A Falbo, Giancarlo Guizzardi, Aldo Gangemi, and Valentina Presutti. Ontology patterns: clarifying concepts and terminology. In *the 4th Workshop on Ontology and Semantic Web Patterns*, 2013.

[160] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE communications magazine*, 44(1):74–82, 2006.

[161] Katrin Bilstrup, Elisabeth Uhlemann, ErikG Ström, and Urban Bilstrup. On the ability of the 802.11 p mac method and stdma to support real-time vehicle-to-vehicle communication. *EURASIP Journal on Wireless Communications and Networking*, 2009(1):902414, 2009.

[162] Li Jiang, Da-You Liu, and Bo Yang. Smart home research. In *IEEE Machine Learning and Cybernetics*, volume 2, pages 659–663, 2004.

[163] Vincent Ricquebourg, David Menga, David Durand, Bruno Marhic, Laurent Delahoche, and Christophe Loge. The smart home concept: our immediate future. In *IEEE Int. Conf. on E-Learn. Ind. Electron*, pages 23–28, 2006.

[164] James J Spilker Jr. Digital communications by satellite. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1977. 685 p.*, 1977.

[165] Gérard Maral and Michel Bousquet. *Satellite communications systems: systems, techniques and technology.* John Wiley & Sons, 2011.

[166] Georgios V Gkoutos, Paul N Schofield, and Robert Hoehndorf. The units ontology: a tool for integrating units of measurement in science. *Database*, 2012:bas033, 2012.

[167] Qianru Zhou, Alasdair J. G. Gray, and Stephen McLaughlin. Seanet: Semantic enabled autonomic management of software defined networks. manuscript in preparation.

[168] Qianru Zhou, Alasdair J. G. Gray, and Stephen McLaughlin. Sara: Semantic access point resource allocation service. manuscript in preparation.

[169] Charalampos Rotsos, Arsham Farshad, Daniel King, David Hutchison, Qianru Zhou, Alasdair JG Gray, Cheng-Xiang Wang, and Stephen McLaughlin. Reasonet: Inferring network policies using ontologies. 2018.

[170] Ramon R Fontes, Samira Afzal, Samuel HB Brito, Mateus AS Santos, and Christian Esteve Rothenberg. Mininet-wifi: Emulating software-defined wireless networks. In *IEEE CNSM*, pages 384–389, 2015.

[171] Chadi Barakat, Eitan Altman, and Walid Dabbous. On tcp performance in a heterogeneous network: a survey. *IEEE Communications Magazine*, 38(1):40–46, 2000.

[172] Olga Ormond, Philip Perry, and John Murphy. Network selection decision in wireless heterogeneous networks. In *IEEE PiMRC*, volume 4, pages 2680–2684, 2005.

[173] Li Qing, Qingxin Zhu, and Mingwen Wang. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Computer communications*, 29(12):2230–2237, 2006.

[174] Suman Sarkar, Sudhir Dixit, and Biswanath Mukherjee. Hybrid wireless-optical broadband-access network (woban): A review of relevant challenges. *Journal of lightwave technology*, 25(11):3329–3340, 2007.

[175] Anthony J Nicholson, Yatin Chawathe, Mike Y Chen, Brian D Noble, and David Wetherall. Improved access point selection. In *ACM MobiSys*, pages 233–245, 2006.

[176] Glenn Judd and Peter Steenkiste. Fixing 802. 11 access point selection. *ACM SIGCOMM Computer Communication Review*, 32(3):31–31, 2002.

[177] Kimaya Mittal, Elizabeth M Belding, and Subhash Suri. A game-theoretic analysis of wireless access point selection by mobile users. *Computer Communications*, 31(10):2049–2062, 2008.

[178] Lin Chen. A distributed access point selection algorithm based on no-regret learning for wireless access networks. In *IEEE VTC*, pages 1–5, 2010.

[179] Sudarshan Vasudevan, Konstantina Papagiannaki, Christophe Diot, Jim Kurose, and Don Towsley. Facilitating access point selection in ieee 802.11 wireless networks. In *ACM SIGCOMM*, pages 26–26. Usenix Association, 2005.

[180] Yutaka Fukuda, Takamitsu Abe, and Yuji Oie. Decentralized access point selection architecture for wireless lans. In *Wireless Telecommunications Symposium*, volume 2004, pages 137–145, 2004.

[181] Goshi Sato, Noriki Uchida, and Yoshitaka Shibata. Resilient disaster network based on software defined cognitive wireless network technology. *Mobile Information Systems*, 2015, 2015.

[182] Xiping Wu, Dushyantha Basnayaka, Majid Safari, and Harald Haas. Two-stage access point selection for hybrid vlc and rf networks. In *IEEE PIMRC*, pages 1–6. IEEE, 2016.

[183] Xiaohui Chen, Wei Yuan, Wenqing Cheng, Wei Liu, and Henry Leung. Access point selection under qos requirements in variable channel-width wlans. *IEEE Wireless Communications Letters*, 2(1):114–117, 2013.

[184] Vishnu Navda, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel, and Samir Das. Mobisteer: using steerable beam directional antenna for vehicular network access. In *ACM MobiSys*, pages 192–205, 2007.

[185] Cheng Xiaonan, Liu Minyan, and He Longjun. Shipwreck statistical analysis and suggestions for ships carrying liquefiable solid bulk cargoes in china. *Procedia Engineering*, 84:188–194, 2014.

[186] Hui Xing, Anqin Lv, Zunshan Zhou, Hongliang Yu, and Shulin Duan. Shipwreck accident probabilistic risk assessment based on event tree analysis. *Journal of Dalian Maritime University*, 1:009, 2012.

[187] Eastern Star Shipwreck Accident. Available: https://en.wikipedia.org/wiki/Sinking_of_Dongfang_zhi_Xing.

[188] Shipwreck since 1833. Available: http://www.infoplease.com/ipa.

[189] Peace time disasters. Available: https://en.wikipedia.org/wiki/List_of_maritime_disasters_in_the_21st_century-Peacetime_disasters.

[190] Missing migrates 2017. Available: https://missingmigrants.iom.int.

[191] Svein Willassen. Forensics and the gsm mobile telephone system. *International Journal of Digital Evidence*, 2(1):1–17, 2003.

[192] Emanuele Della Valle, Stefano Ceri, Frank Van Harmelen, and Dieter Fensel. It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6), 2009.

[193] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-sparql: Sparql for continuous querying. In *Proceedings of the 18th international conference on World wide web*, pages 1061–1062. ACM, 2009.

[194] Jean-Paul Calbimonte, Oscar Corcho, and Alasdair JG Gray. Enabling ontology-based access to streaming data sources. In *International Semantic Web Conference*, pages 96–111. Springer, 2010.

[195] Alessandro Margara, Jacopo Urbani, Frank Van Harmelen, and Henri Bal. Streaming the web: Reasoning over dynamic data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 25:24–44, 2014.

[196] Gowrishankar and PS Satyanarayana. Neural network based traffic prediction for wireless data networks. *International Journal of Computational Intelligence Systems*, 1(4):379–389, 2008.

[197] 3GPP LTE specications 2017. Available: http://www.3gpp.org/specifications.

[198] Ministry of industry and information technology of the people's republic of china. 2017. Available: http://www.miit.gov.cn/.

[199] Maxim Kolchin, Dmitry Mouromtsev, and Sergey A Arustamov. Web-based visualisation and monitoring of smart meters using cqels. In *ISWC TC/SSN*, pages 109–112, 2014.

[200] Alessandra Mileo, Ahmed Abdelrahman, Sean Policarpio, and Manfred Hauswirth. Streamrule: a nonmonotonic stream reasoning system for the semantic web. In *International Conference on Web Reasoning and Rule Systems*, pages 247–252. Springer, 2013.

[201] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Chan Le Van, and Manfred Hauswirth. Elastic and scalable processing of linked stream data in the cloud. In *International Semantic Web Conference*, pages 280–297. Springer, 2013.

[202] Jean-Paul Calbimonte, Jose Mora, and Oscar Corcho. Query rewriting in rdf stream processing. In *International Semantic Web Conference*, pages 486–502. Springer, 2016.

[203] Enhanced 911 - wireless services, FCC, 2017. Available: https://www.fcc.gov/general/enhanced-9-1-1-wireless-services.

[204] Fabrizio Antonelli, Matteo Azzi, Marco Balduini, Paolo Ciuccarelli, Emanuele Della Valle, and Roberto Larcher. City sensing: visualising mobile and social data about a city scale event. In *ACM Advanced Visual Interfaces*, pages 337–338. ACM, 2014.

[205] John Steenbruggen, Emmanouil Tranos, and Peter Nijkamp. Data from mobile phone operators: A tool for smarter cities? *Telecommunications Policy*, 39(3-4):335–346, 2015.

[206] Marta C González, Pedro G Lind, and Hans J Herrmann. System of mobile agents to model social networks. *Physical review letters*, 96(8):088702, 2006.

[207] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[208] Kostis Kyzirakos, Manos Karpathiotakis, George Garbis, Charalampos Nikolaou, Konstantina Bereta, Ioannis Papoutsis, Themos Herekakis, Dimitrios Michail, Manolis Koubarakis, and Charalambos Kontoes. Wildfire monitoring using satellite images, ontologies and linked geospatial data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 24:18–26, 2014.

[209] Thomas Eiter, Josiane Xavier Parreira, and Patrik Schneider. Spatial ontology-mediated query answering over mobility streams. In *European Semantic Web Conference*, pages 219–237. Springer, 2017.

[210] Hoan Nguyen Mau Quoc and Danh Le Phuoc. An elastic and scalable spatiotemporal query processing for linked sensor data. In *ACM ICSS*, pages 17–24, 2015.

[211] Krispijn Scholte and Léon JM Rothkrantz. Personal warning system for vessels under bad weather conditions. In *ISCRAM*, 2014.

[212] Alessio Malizia, Francisco Astorga-Paliza, Teresa Onorati, Paloma Díaz Pérez, and Ignacio Aedo. Emergency alerts for all: an ontology based approach to improve accessibility in emergency alerting systems. 2008.

[213] Mauro Brunato and Roberto Battiti. Statistical learning theory for location fingerprinting in wireless lans. *Computer Networks*, 47(6):825–845, 2005.

[214] Joseph Mitola. Cognitive radio—an integrated agent architecture for software defined radio. 2000.

[215] Charles Clancy, Joe Hecker, Erich Stuntebeck, and Tim O'Shea. Applications of machine learning to cognitive radio networks. *IEEE Wireless Communications*, 14(4), 2007.

[216] Jeremy Roschelle. Keynote paper: Unlocking the learning value of wireless mobile devices. *Journal of computer assisted learning*, 19(3):260–272, 2003.

[217] Sebastian Zander, Thuy Nguyen, and Grenville Armitage. Automated traffic classification and application identification using machine learning. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 250–257. IEEE, 2005.

[218] Thomas G Dietterich, Pedro Domingos, Lise Getoor, Stephen Muggleton, and Prasad Tadepalli. Structured machine learning: the next ten years. *Machine Learning*, 73(1):3, 2008.

[219] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[220] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[221] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[222] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *ACL*, pages 79–86. Association for Computational Linguistics, 2002.

[223] Christian Robert. Machine learning, a probabilistic perspective, 2014.

[224] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[225] David J Hand. Principles of data mining. *Drug safety*, 30(7):621–622, 2007.

[226] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in knowledge discovery and data mining*, volume 21. AAAI press Menlo Park, 1996.

[227] The state of data science & machine learning. Available: "https://www.kaggle.com/surveys/2017?utm=cade".

[228] Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. Lodstats: The data web census dataset. In *International Semantic Web Conference*, pages 38–46. Springer, 2016.

[229] Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920, 2017.

[230] Jens Lehmann, Gezim Sejdiu, Lorenz Bühmann, Patrick Westphal, Claus Stadler, Ivan Ermilov, Simon Bin, Nilesh Chakraborty, Muhammad Saleem, Axel-Cyrille Ngonga Ngomo, and Hajira Jabeen. Distributed semantic analytics using the sansa stack. In *International Semantic Web Conference*, pages 147–155. Springer, 2017.

[231] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. Neural network-based question answering over knowledge graphs on word and character level. In *WWW*, pages 1211–1220, 2017.

[232] Patrick Westphal, Lorenz Bühmann, Simon Bin, Hajira Jabeen, and Jens Lehmann. Sml-bench–a benchmarking framework for structured machine learning. 2017.