International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3320

(08/2014)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Future networks

Requirements for applying formal methods to software-defined networking

Recommendation ITU-T Y.3320

ITU-T

International Telecommunication Union

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	**Y.3000–Y.3499**
CLOUD COMPUTING	Y.3500–Y.3999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3320

Requirements for applying formal methods to software-defined networking

Summary

Recommendation ITU-T Y.3320 provides a descriptive overview and requirements for applying formal methods to software-defined networking (SDN).

Appendix I, introduces an example demonstrating how formal methods are applied to SDN environments.

Formal methods are mathematics-based techniques used for specifying, developing, and verifying software and hardware systems and are expected to increase the reliability and robustness of the system. In SDN environments the consistency, reliability and security of applications are important as incomplete or malicious programmable entities could cause a break-down of underlying networks. In this sense, the use of formal methods can be an effective approach to the mitigation of such problems.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3320	2014-08-29	13	11.1002/1000/12284

Keywords

Formal methods, formal specification, formal verification, future network, software defined networking.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3320

Requirements for applying formal methods to software-defined networking

1 Scope

This Recommendation describes requirements for using formal methods. Formal methods are mathematics-based techniques used to specify, develop and verify software and hardware systems in the context of software-defined networking (SDN) for future networks (FN).

The scope of this Recommendation covers:

- an overview of formal methods (formal specification and formal verification) for SDN, and
- requirements for applying formal methods to SDN.

An example of how to apply formal methods to SDN is provided in Appendix I.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3001] Recommendation ITU-T Y.3001 (2011), *Future networks: Objectives and design goals.*

[ITU-T Y.3011] Recommendation ITU-T Y.3011 (2012), *Framework of network virtualization for future networks.*

[ITU-T Y.3300] *Recommendation* ITU-T Y.3300 (2014), *Framework of software-defined networking.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 future network [ITU-T Y.3001]: A network able to provide services, capabilities and facilities difficult to provide using existing network technologies. A future network is either:

a) A new component network or an enhanced version of an existing one, or

b) A heterogeneous collection of new component networks or of new and existing component networks that is operated as a single network.

3.1.2 network virtualization [ITU-T Y.3011]: A technology that enables the creation of logically isolated network partitions over shared physical networks so that heterogeneous collections of multiple virtual networks can simultaneously coexist over the shared networks. This includes the aggregation of multiple resources in a provider and appearing as a single resource.

3.1.3 software-defined networking [ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

FN Future Network

SDN Software-Defined Networking

5 Conventions

In this Recommendation, the following conventions are used:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Introduction

Formal methods are software engineering techniques based on the mathematical representation and analysis of software programs and/or hardware. They include formal specification, analysis of specification and formal verification of software and/or hardware behaviour [b-Clarke]. The formal specification describes the semantics of a system using mathematical representation. The formal verification is the act of proving or disproving the correctness of designs or implementations with respect to the formal specification.

Objectives and design goals for future networks are described in [ITU-T Y.3001] which also identifies the high-level capabilities and characteristics that need to be supported by such future networks. To design and implement systems that conform to the design goals identified in [ITU-T Y.3001], the structure and behaviour of the systems need to be described in a way that prevents misinterpretation of the intended meanings and that avoids inconsistency in the systems. FNs may be used for mission critical systems and/or other areas, such as private clouds and data centers, where systems have significant requirements for reliability and consistency and where otherwise a catastrophic disaster could occur.

[ITU-T Y.3300] describes the framework and fundamentals of SDN which it describes as a new networking approach that enables network resources to be directly programed, orchestrated and controlled.

Finally [ITU-T Y.3011] describes network abstraction into networks that manage the virtualization of the networks.

SDN facilitates network operators to introduce new capabilities by writing simple software programs that control the network in a programmable way. In SDN based networks it is important to check the consistency and security properties of these programs before their installation or deployment in the network.

As described previously, an effective approach to increasing reliability and reducing any misunderstanding or inconsistency of the meaning of the systems or programs is the adoption of formal methods. Formal methods help network operators and application developers protect their systems from unexpected errors that might not be caught during the development and deployment process.

7 Overview of applying formal methods to software-defined networking

Figure 1 illustrates the overall flow from the informal description of the conceptual model, logical design and domain specific network through to the implementation of a system using formal methods.

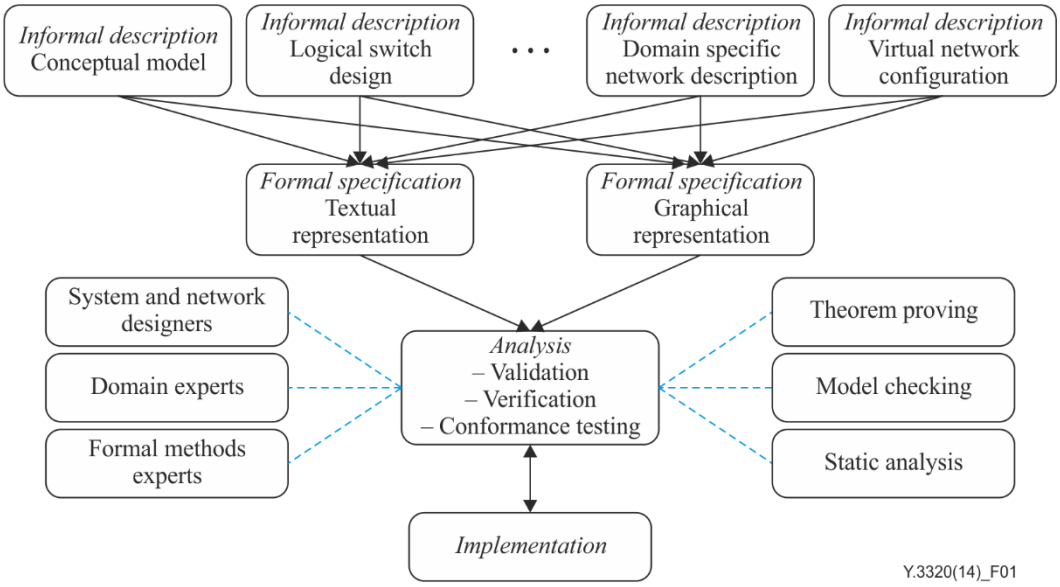


Figure 1 – Overall flow of formal methods

Traditional methods for realizing network protocols and devices are based on community agreements on informal descriptions of such mechanisms [b-Griffin]. As depicted in Figure 1, these traditional methods can be improved by applying formal methods to the development process of software-defined networking [b-Wang]. Targets of specification can conceptually model components or mechanisms, logical switch/router models, network protocols, topologies of virtual networks and so on. Informal descriptions of those targets can be encoded in formal specification languages that can reflect the features of targets of the existing methods. The formal specification can be in textual or graphical form to represent their semantics. Once the specifications are described formally, system and network designers can check for the existence of inconsistencies and possible errors in the specification with the help of formal methods experts and/or supporting tools. Various types of formal verification methods (e.g., theorem proving, model checking and static analysis) can be applied to the validation and verification process.

The high-level architecture of SDN consists of four layers as defined in [ITU-T Y.3300]:

– Application layer
– SDN control layer
– Resource layer
– Management functions multi-layer

In order to provide formal methods to SDN, additional functions comprised in the formal methods component are needed to process the formal specification and the formal verification, (see Figure 2). The formal specification function provides an application-formal methods interface for

SDN applications to express their semantics in a formal representation. The formal verification function provides a control-formal methods interface for the SDN controller to check whether new semantics can cause errors or operational conflicts within the network.

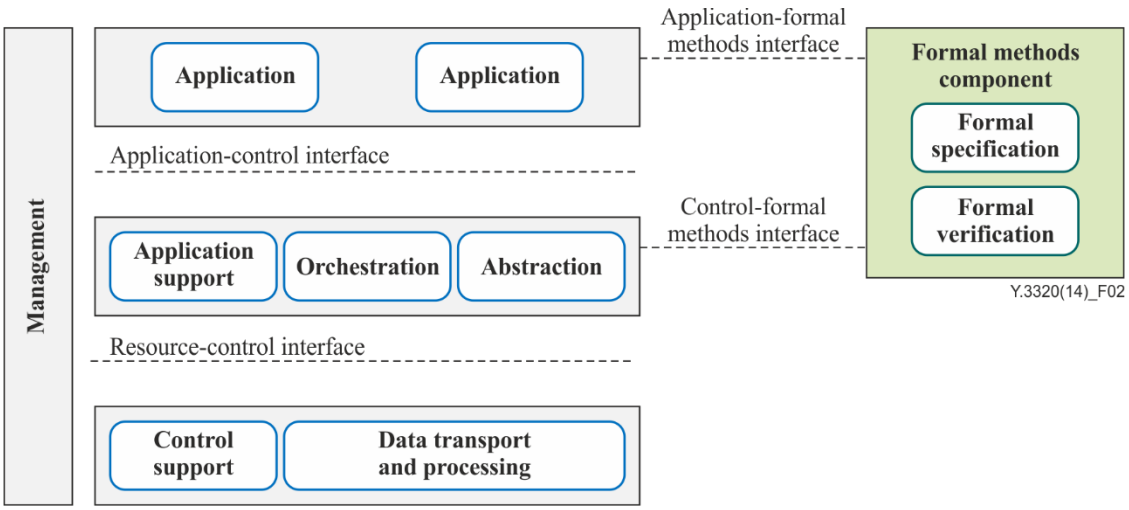
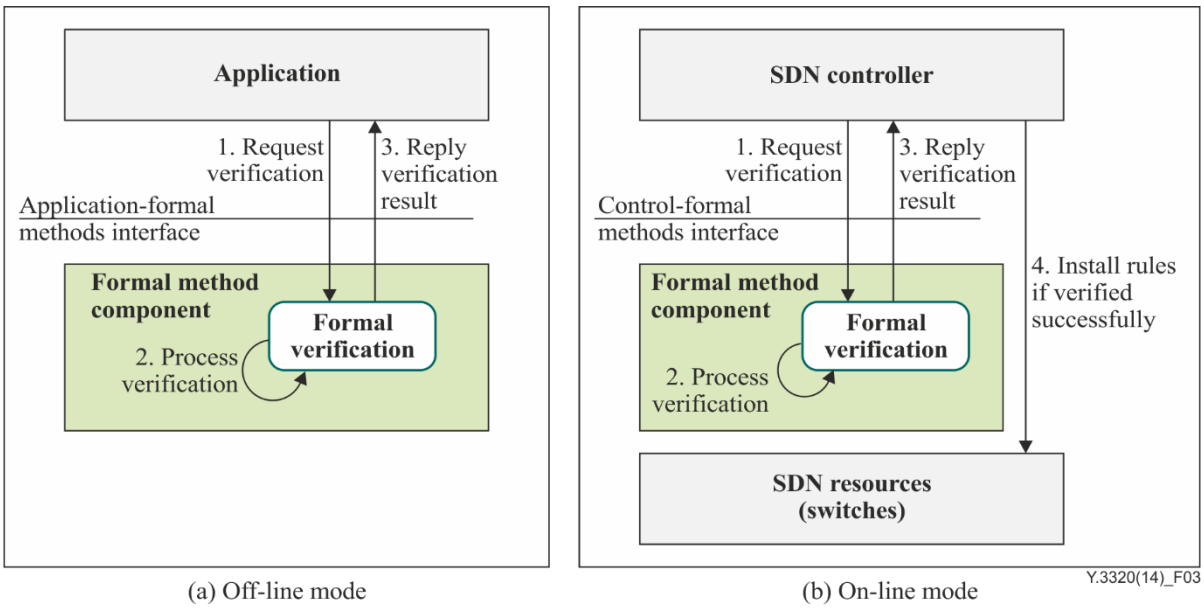


Figure 2 – Relationship between the formal methods component and the high-level SDN architecture

The formal methods component supports different operation modes which correspond to off-line operation mode and on-line operation mode. In the off-line operation mode, the formal methods component receives a request from applications via the application-formal methods interface and performs the necessary functions to process the specification or verification without interaction with the SDN controller. In the on-line operation mode, the formal methods component receives a request from the SDN controller via the control-formal methods interface during the runtime of the network.

Figure 3 illustrates operational procedures of these different modes.



(a) Off-line mode (b) On-line mode

Figure 3 – Off-line and on-line modes of formal verification

8 Functional requirements

This clause identifies a set of functional requirements that must be supported by the formal methods component.

8.1 General requirements (GR)

GR-1: It is recommended that the formal methods component guarantees that the design and implementation of programmable network resources conforms to the standards, both in terms of correctness and in respect of security properties.

NOTE – Programmable network resources can be customized in software or hardware, or in a combination of both, depending on the requirements from service providers, application/service developers and network operators to produce an optimal solution for their uses.

GR-2: It is required that the formal methods component provides open interfaces to interact with the application and/or SDN controller.

NOTE 1 – The application and/or SDN controller sends requests to use formal specification and/or verification functions provided by the formal methods component.

NOTE 2 – After performing the specification or verification process, the formal methods component sends results back to the application and SDN controller.

GR-3: It is recommended that the formal methods component supports policies or rules required by applications in heterogeneous network environments.

NOTE – Control and management entities that span across SDN-based networks which are owned by different stake-holders, management authorities or vendors, need to be controlled in distributed environments according to nature. Multiple entities may need to be coordinated to conform to the higher level requirements of heterogeneous networks.

GR-4: It is recommended that the formal methods component supports authentication and authorization to confirm the identity and access rights of users.

8.2 Requirements of formal specification (FS)

FS-1: It is recommended that the formal methods component supports formal syntax and semantics in high-level languages, APIs and underlying protocols for SDN.

NOTE 1 – High-level languages that interface with control and management entities need to have formal semantics to avoid any confusion in the interpretation of the underlying mechanism and/or network configuration.

NOTE 2 – Specifications of programmable network devices such as switches and protocols for controlling those devices need to be proved safe and consistent to provide a stable foundation for the SDN application and services.

NOTE 3 – Properties that need to be satisfied with SDN should be described in notations with formal semantics.

NOTE 4 – An action or a set of actions associated with target properties must be described in notations with formal semantics. Examples of actions include packet forwarding, packet modifications and group table or pipeline processing.

FS-2: It is recommended that the formal methods component supports a conceptual model to reason properties and behaviors about networks defined, configured and/or implemented by software and/or hardware for SDN.

8.3 Requirements of formal verification (FV)

8.3.1 Consistency and security

FV-1: It is required that the formal methods component checks the consistency and security of network configurations, the virtual/physical topologies and the intended properties of the network resources.

NOTE 1 – Examples of intended properties include the following:

- No routing loops and/or non-reachable points in the network
- No conflicts between logical and physical networking resource assignment for applications
- No conflicts in dynamic network update where new or update configurations conform to properties of the network and do not break the consistency of existing networks

FV-2: It is required that the formal methods component checks the consistency of the application and policies against conflicts that can occur between SDN entities or in network configurations.

NOTE – Examples of conflicts include the following:

- Rule or behaviour conflicts between multiple applications in a controller.

8.3.2 Operations

FV-3: It is recommended that the formal methods component supports different verification modes (e.g., off-line mode and on-line mode) and different verification methodologies (e.g., symbolic or non-symbolic manners).

FV-4: For the symbolic verification methodology, it is recommended that the formal methods component supports symbolic model checking technologies in order to verify whether the application meets the specification.

FV-5: The formal verification component can optionally support the following operations.

- Collecting the information of network topology and data forwarding paths (e.g., a flow table in OpenFlow [b-ONF]) from an SDN controller
- Creating description code in a predefined formal language based on the collected information and generating a symbolic transition graph using the created description code in the formal language
- Applying a verification operation to the created transition graph
- Providing verification results in the formula based on the Boolean expression (e.g., binary decision diagram or conjunctive normal form)

8.3.3 Resource information discovery

FV-6: It is required that the formal methods component obtains the information on network resources and SDN entities including static information (e.g., network topology and capability of SDN entities) and dynamic information (e.g., re-configured topology and status of SDN entities).

FV-7: It is recommended that the formal methods component obtains the entire network topology information within an administrative domain from the SDN controller(s).

8.4 Miscellaneous

Mic-1: The formal methods component can optionally provide statistics about the verification process such as a summary of verification results, processing time and other performance data.

9 Environmental considerations

Applying formal methods to SDN reduces any inconsistency or ambiguity of SDN applications by enabling the application to be specified and verified before an execution in the network. It can prevent malfunctions, misuses and unexpected behaviour of SDN applications and so should optimize the resource usage, which reduces energy consumption.

However, the verification process requires additional computational resources to examine SDN applications in the network and energy consumption may increase.

10 Security considerations

SDN facilitates network operator's control of their networks in an automatic and programmable way. In other words, SDN network operators including application developers can introduce a new capability by writing a program. Many important properties such as programmability, reliability, flexibility and customization of network resources, need to be supported by SDN. However, these properties can cause unexpected security problems in SDN environments as incomplete or malicious programs can also be introduced easily into the networks. Consequently, programs need to be thoroughly examined before their installation to the networks. For SDN therefore, the use of formal methods can be an effective approach to checking that programs are operating correctly and avoiding possible inconsistency or misinterpretation of their networking behaviours. Security issues therefore should be considered and resolved during the formal specification and verification stage.

Appendix I

Overview of formal methods for networking

(This appendix does not form an integral part of this Recommendation.)

The objective of this appendix is to provide an example showing how a formal specification and verification process can be applied to SDN. In SDN environments consistency and security are important to help ensure that errors or operational conflicts are not introduced into the networks. In this sense, formal methods are effective in the specification and verification of applications or programs in order to avoid inconsistency or misinterpretation of their networking behaviours.

In order to avoid implementation dependency in describing the example used in this appendix, logical functional descriptions are used to describe behaviours and operations related to formal specification and verification.

I.1 High level operational model of SDN and formal methods tool

Figure I.1 illustrates a high level operational model of SDN including the formal methods tool that provides the specification syntax and semantics for applications and the verification procedures to check their operational behaviours within the networks.

The following assumptions are considered in this example [b-Canini].

– It is an OpenFlow based SDN environment where the OpenFlow protocol (version 1.0 ~ 1.3) is used for communication between the controller and switches.

– The controller manages the OpenFlow based global network information and it needs to provide this global network information when it receives a request from the formal methods tool.

– The formal description code considered in this section is based on the fields defined in the OpenFlow protocol.

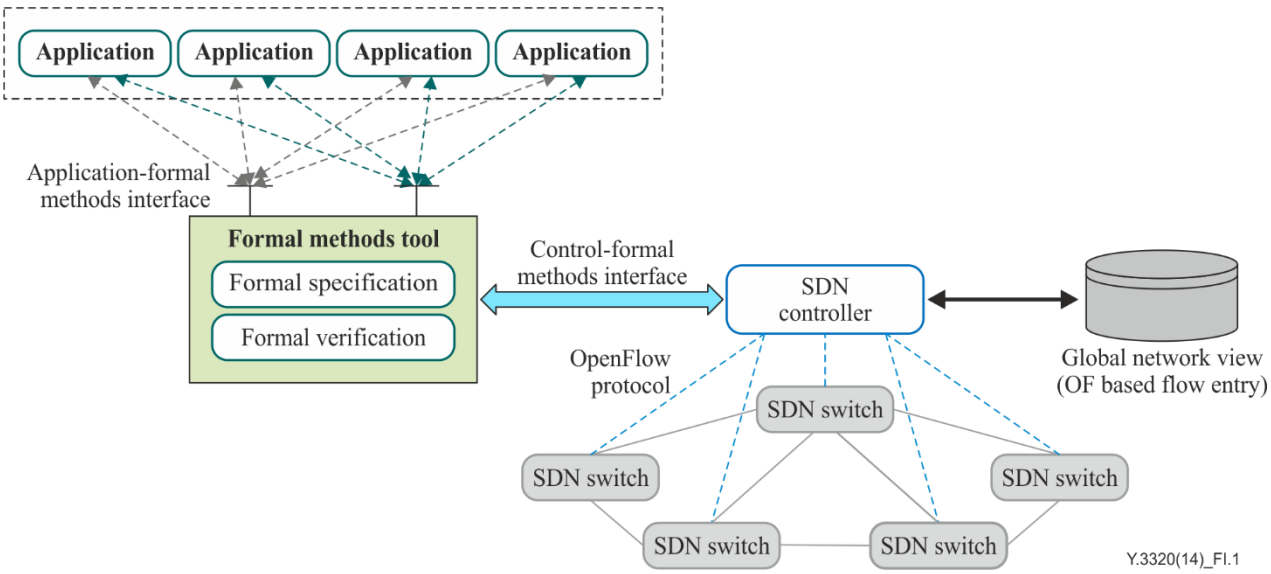


Figure I.1 – High level operational model

I.2 Formal specification tool

Applications need to express their policies or rules in a formal specification supported by the formal methods tool. The formal specification is a representation expressed in a language whose semantics are formally defined based on logics and mathematics (i.e., it is not a natural language).

Figure I.2 shows the interaction between the application and the formal methods tool [b-Shin]. The application uses the application programming interface (API) of the formal specification to request the conversion of their rules and receive formal description code as a result.

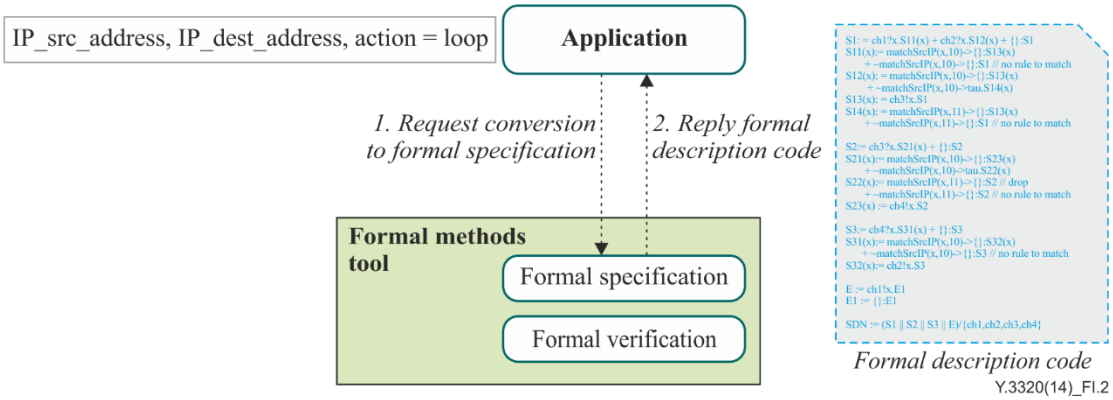


Figure I.2 – Formal specification tool with interfaces

Figure I.3 illustrates in greater detail how the formal specification is processed. When the formal specification receives the request, it gathers the application policy and the global network information from the controller. Then it converts these inputs into a formal description code using the formal language defined. Finally, it returns the formal description code to the application.

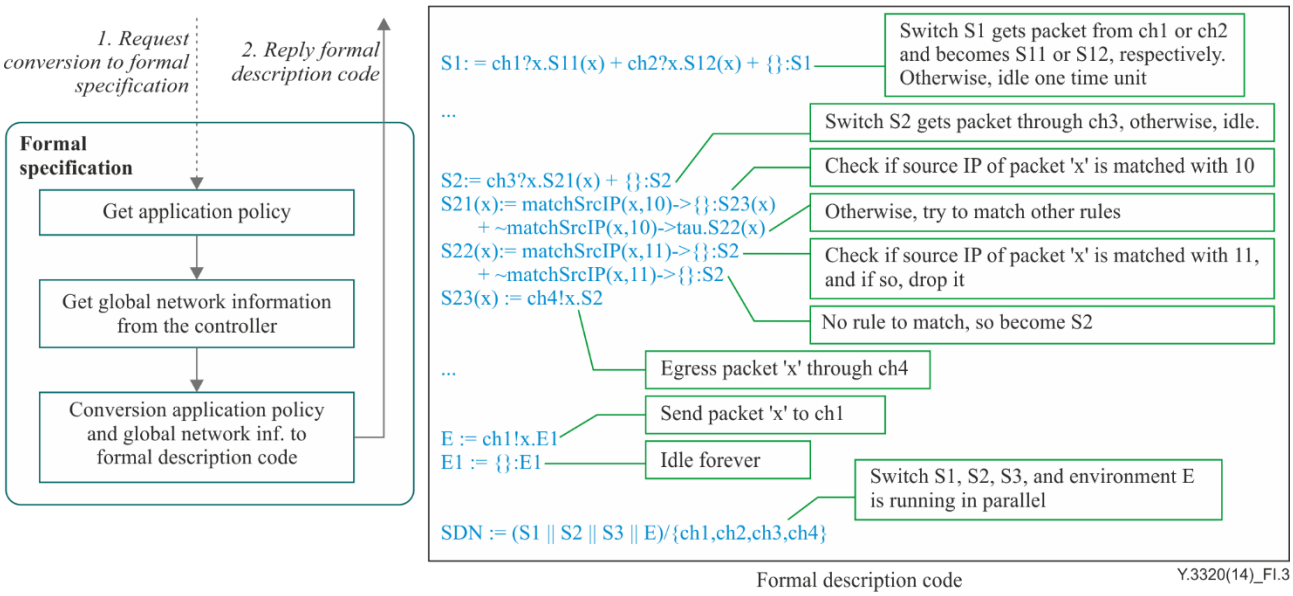


Figure I.3 – Specification procedures and formal description code

I.3 Formal verification tool

When the application gets the formal description code, it can request the verification tool to check whether any errors or operational conflicts may occur within the networks. Figure I.4 shows the verification procedures in more detail. When the formal verification receives the request, it parses the formal description code received, builds a transition graph and starts to traverse the transition graph created previously to check the consistency or correctness with respect to the application rules and the global network operations.

Figure I.5 shows an example of a transition graph whose construction was based on the formal description code. The transition graph is a directed graph where boolean, action and assignments are labelled [b-Shin]. From each state, when boolean holds, action and assignments are performed and

the current state proceeds to the next state. In this transition graph, each field in a flow entry and a packet can be represented symbolically as a binary decision diagram or as a conjunctive normal form. In order to verify the existence of a loop in this example, it can apply a first search starting from the root state "S1||S2||S3" and identify every loop. To find the loops that a packet is looping, actions on the edge should be detected based on the rules defined for the loop detection. The red line in Figure I.5 indicates that looping may occur in the networks. Finally, the formal verification tool sends back the verification results to the application.

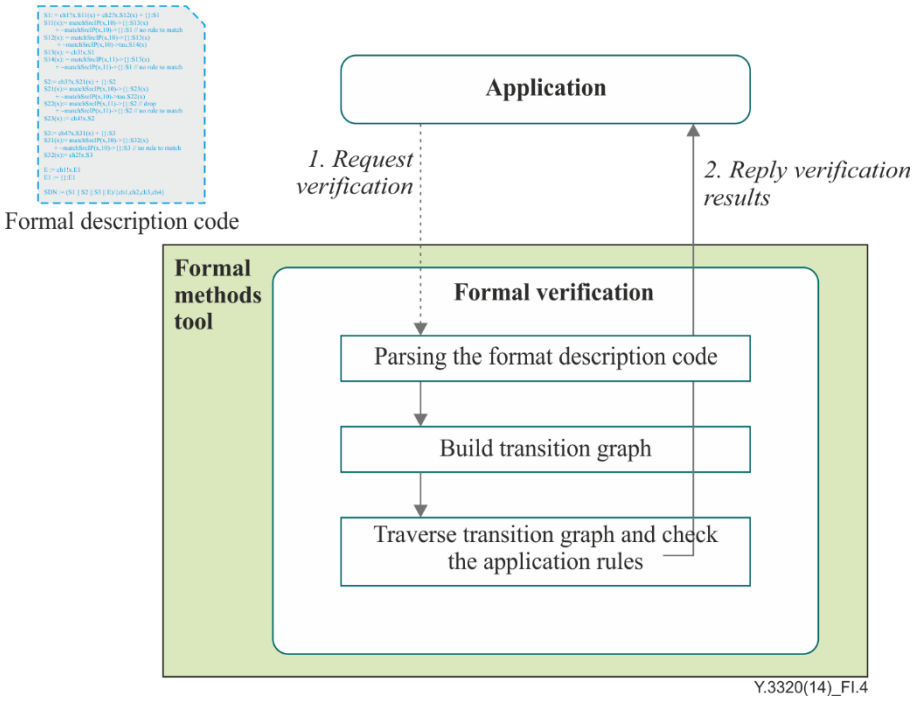


Figure I.4 – Verification procedures

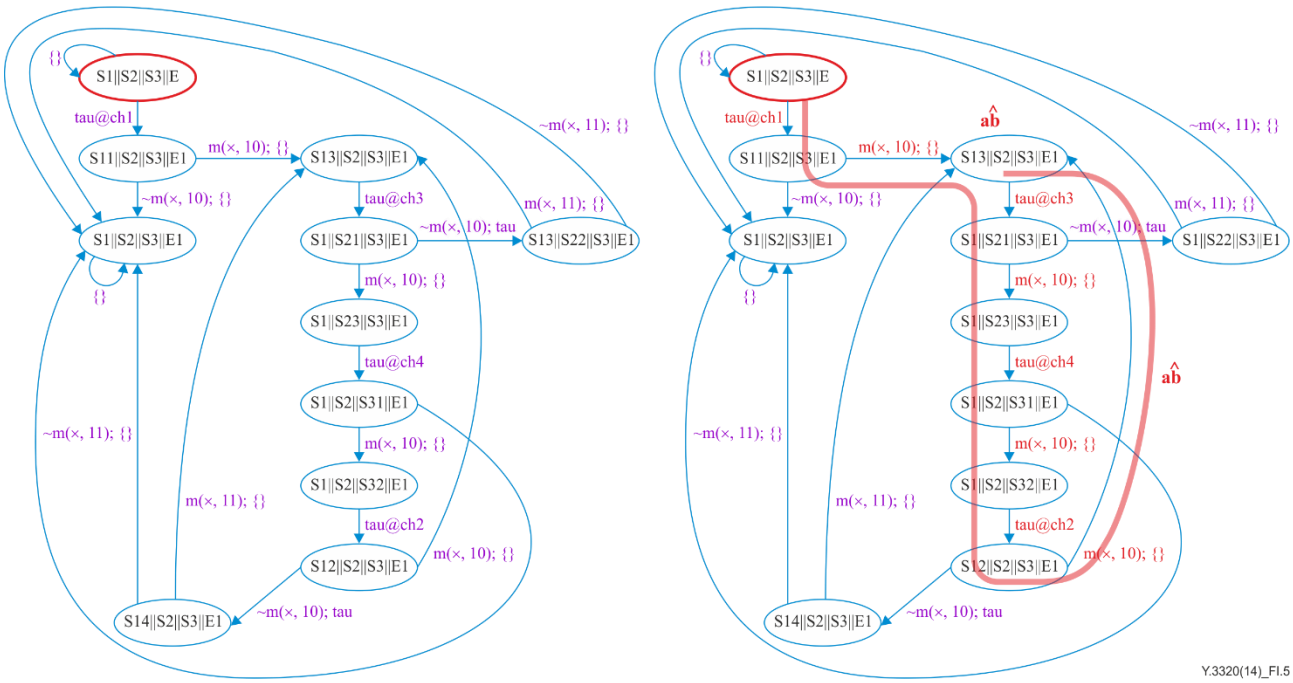


Figure I.5 – Transition graph and traverse results

Bibliography

[b-Bishop] Steve Bishop, *et. al.* (2005), *Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and Sockets*. http://conferences.sigcomm.org/sigcomm/2005/paper-BisFai.pdf .

[b-Canini] Marco Canini, *et. al*. (2012), *A NICE way to test OpenFlow applications*. http://infoscience.epfl.ch/record/170618/files/nsdi-final

[b-Clarke] Edmund M. Clarke and Jeannette M. Wing (1996), "*Formal methods: state of the art and future directions*". https://www.site.uottawa.ca/~afelty/csi5110/state_art_future.pdf .

[b-Griffin] Timothy G. Griffin, *Do Formal Methodists have Bell-Shaped Heads?* Proceedings of the First Workshop on Automated Theory Engineering.

[b-ONF] Open Networking Foundation, *OpenFlow/Software-Defined Networking (SDN)*, https://www.opennetworking.org/

[b-Shin] M. Shin, H. Kwak, *et. al.* (2013), *Process Algebra Based Symbolic Verification Framework for Software-Defined Networking.* Telecommunications Review Vol. 23 No. 5.

[b-Wang] Anduo Wang, *et. al.* (2011), *FSR: Formal analysis and implementation toolkit for safe interdomain routing*, ACM SIGCOMM Conference on Data Communication.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2015