Software Impacts 1 (2019) 100003

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

IRATI: Open source RINA implementation for Linux

Eduard Grasa [a,*], Miquel Tarzan [a], Leonardo Bergesio [a], Bernat Gastón [a], Vincenzo Maffione [b], Francesco Salvestrini [b], Sander Vrijders [c], Dimitri Staessens [c]

[a] Fundació i2CAT, C/Gran Capità 2-4, Edifici Nexus I, Oficina 203, Barcelona, Spain
[b] Nextworks, Via Livornese, 56122 San Piero A Grado Pl, Italy
[c] Ghent University – imec Technologiepark-Zwijnaarde 15, 9052 Ghent, Belgium

ARTICLE INFO

Keywords:
Network architecture
RINA
Future internet experimentation
Network protocols
Network virtualization
Network programmability

ABSTRACT

IRATI is a open source implementation of RINA for OS/Linux systems that allows researchers and innovators to experiment with RINA networks. RINA is a new Internetwork architecture that supports without the need of extra mechanisms mobility, multi-homing and Quality of Service, provides a secure and configurable environment and allows for a seamless adoption. IRATI implements the core RINA protocols and multiple policies to customize such protocols to different environments. Policies can be developed via a Software Development Kit (SDK). IRATI provides an API for applications to natively use RINA IPC services, as well as multiple monitoring utilities and example applications.

Code metadata

Current Code version	*2.3.0*
Permanent link to code / repository used of this code version	https://github.com/SoftwareImpacts/SIMPAC-2019-4
Legal Code License	*GPL, LGPL*
Code Versioning system used	*git*
Software Code Language used	*C, C++*
Compilation requirements, Operating environments & dependencies	https://github.com/IRATI/stack/tree/2.3.0#2-build-instructions
If available Link to developer documentation / manual	https://github.com/IRATI/stack/wiki/IRATI-in-depth
Support email for questions	irati@freelists.org

Software metadata

Current software version	*2.3.0*
Permanent link to executables of this version	https://github.com/IRATI/stack/releases/tag/2.3.0
Legal Software License	*GPL, LGPL*
Computing platform / Operating System	*Linux kernel 4.x, tested distros: Debian 8, Debian 9, Arch Linux, Raspbian 4.9, Ubuntu 16.04*
Installation requirements & dependencies	https://github.com/IRATI/stack/tree/2.3.0#2-build-instructions
If available Link to user manual — if formally published include a reference to the publication in the reference list	https://github.com/IRATI/stack/tree/2.3.0#1-introduction
Support email for questions	irati@freelists.org

RINA overview

During the last decade research funding bodies have allocated money to fund "Future Internet" [1] or "Clean-slate" designs that could reduce network complexity by redesigning the network protocol architecture, questioning some of its key principles. Industry groups such as the ETSI ISG NGP [2] are looking at alternatives to the current "TCP/IP" protocol suite. However, few initiatives have really been able to "clean the slate" and question the core model and underlying

* Corresponding author.
E-mail addresses: eduard.grasa@i2cat.net (E. Grasa), miquel.tarzan@i2cat.net (M. Tarzan), leonardo.bergesio@i2cat.net (L. Bergesio), bernat.gaston@i2cat.net (B. Gastón), v.maffione@nextworks.it (V. Maffione), f.salvestrini@nextworks.it (F. Salvestrini), sander.vrijders@ugent.be (S. Vrijders), dimitri.staessens@ugent.be (D. Staessens).

https://doi.org/10.1016/j.simpa.2019.100003
Received 13 June 2019; Received in revised form 8 July 2019; Accepted 15 July 2019

principles of current Internet protocols. Of those who have done it, RINA – the Recursive InterNetwork Architecture [3,4] – is probably the simplest yet most general solution.

RINA goes back to the early days of network research, in which operating systems and distributed applications were the model for thinking about packet networks. Networking was seen as the enabler of distributed computing. Hence the main function of networks was to communicate applications, not devices. RINA builds on the premise that networking is Inter-Process Communication (IPC) and only IPC, to provide a framework that reconstructs the overall structure of the Internet, forming a model that comprises a single repeating layer, the DIF (Distributed IPC Facility), which is the minimal set of components required to allow distributed IPC between application processes (Fig. 1). RINA supports without the need of extra mechanisms mobility [5], multi-homing [6] and Quality of Service, provides a secure and configurable environment [7], motivates for a more competitive marketplace and allows for a seamless adoption.

To date RINA is the only initiative that is proposing a network architecture that can be applied from the application layer down to the physical wire, fixing the shortcomings of current network architectures [8] while at the same time pursuing a minimalistic approach and theory of operation. RINA is based on a single type of repeating layer – the DIF – that features two core protocol frameworks (one for data transfer and another one for layer management) [2] which can be customized via policies to any type of networking environment [9]. The advantages of RINA have been published in a number of industry reports [2] and scientific papers. Such advantages include:

- Drastic simplification of the network architecture, minimizing the number of protocols required to build and operate networks [2, 10,11]; thus minimizing both capital and operational expenditures.
- Greatly simplifies the management of multi-homing and distributed mobility [5], and enables dynamic network renumbering without impacting existing flows [6].
- Provides a greater level of network security at a lower cost compared to current TCP/IP-based networks [7], with greater flexibility and reusability of security policies [12].
- Provides better performance through and effective framework for dealing with congestion management [13], resource allocation and Quality of Service [14] at any network scale.
- Can be deployed over, below, or next to existing network technologies and protocols [2], thus supporting both greenfield and brownfield network deployments [15–17].

Why IRATI?

RINA is a very promising, radically simple network architecture that solves many of the shortcomings of current communications networks, but:

- The core RINA specifications are not one hundred percent ready to use in operational scenarios. There is a need for implementing such specifications, so that they can be debugged and completed.
- Theoretical properties of RINA networks have to be validated via experimental results in a variety of platforms.
- More researchers need to be involved in RINA research and development activities, to create a critical mass that can explore the multiple benefits of RINA in different networking areas and raise the interest of industry. To facilitate this a number of open source RINA implementations and tools need to be developed, so that it becomes easier to start experimenting with RINA networks.
- In order to open a market for RINA-based products and solutions, there is a need to lower the risk of RINA technology adoption, facilitating the process of designing and building products and services that leverage RINA.

Hence, developing a RINA implementation from scratch before IRATI was a difficult task, that could be summarized in two main issues. The first one was the lack of completeness of many RINA specifications, especially those related to RINA's data transfer protocol (called EFCP, Error and Flow Control Protocol). A RINA implementer had to work out with researchers to find out ways to fill in the gaps in the specifications. The second obstacle for a generic, extensible RINA implementation was the nature of RINA per se: since it is a new architecture that addresses all aspects of networking, developing the base system framework requires a significant number of resources (in contrast to networking projects that just extend the TCP/IP protocol suite).

The IRATI RINA implementation [18] is designed to address the problems described by the previous bullet points. IRATI has received the support of the European Commission FP7 and H2020 research funding programs, under the umbrella of three projects:

- *FP7 IRATI* (2013–2014), Investigating RINA as an Alternative to TCP/IP [19]. FP7 IRATI funded the initial development of the IRATI RINA implementation, as well as initial results on experimentation with RINA in the enterprise VPN space.
- *FP7 PRISTINE* (2014–2016), Programmability in RINA for European Supremacy of Virtualized Networks [20]. The project funded the design and development of a Software Development Kit (SDK) to facilitate programmability of policies for different DIFs using the IRATI RINA implementation. It also carried out experimental activities in the areas of datacentre networking, global network overlays and Network Function Virtualisation (NFV).
- *H2020 ARCFIRE* (2016–2018), Large Scale RINA benchmark on FIRE [21]. ARCFIRE provided funding to enhance the usability and maturity of IRATI, making it easier to use for other researchers as well as industry innovators. Experimentation with IRATI focused on RINA for 5G networks, analysing the benefits in network management, resiliency, quality of service, renumbering and mobility.

The work carried out in these projects led to a general-purpose, open-source, comprehensive RINA implementation that provides a worked example of how a RINA system can be implemented for Linux hosts. Its open-source nature allows the reuse of some of its software modules for other implementations. In the course of developing IRATI, most of the critical gaps in the core RINA specifications were completed, drastically facilitating the development of alternative implementations. Moreover, a number of auxiliary tools that facilitate the deployment and testing of RINA implementations were developed as companion projects to IRATI. Examples of these tools are configurators [22], management and orchestration systems for RINA networks [23], or applications ported to the RINA API [24].

What does IRATI do?

IRATI allows a Linux device to behave as a RINA-enabled host or as a RINA software router, enabling a large variety of experimental scenarios. IRATI implements the core protocols of RINA and a number of policies that allow researchers to try different approaches to authentication, encryption, access control, routing, congestion control, forwarding or scheduling. A whole list of the policies already supported by IRATI and its configuration details can be found at [25]. In addition to this, new policies can be easily implemented and deployed via a Software Development Kit (SDK) described in [9].

With IRATI, researchers and innovators can try RINA experiments over Ethernet (with or without VLANs), WiFi, UDP, TCP and shared memory between KVM/QEMU and Xen Virtual Machines and their hosts [26]. It is also possible to deploy IP over RINA [16], enabling scenarios where RINA is used as a transport solution for IP traffic. The tools shipped by IRATI allow monitoring a number of metrics to characterize the behaviour of RINA flows, such as goodput, delay or packet loss. In addition to the information exported via sysfs and the

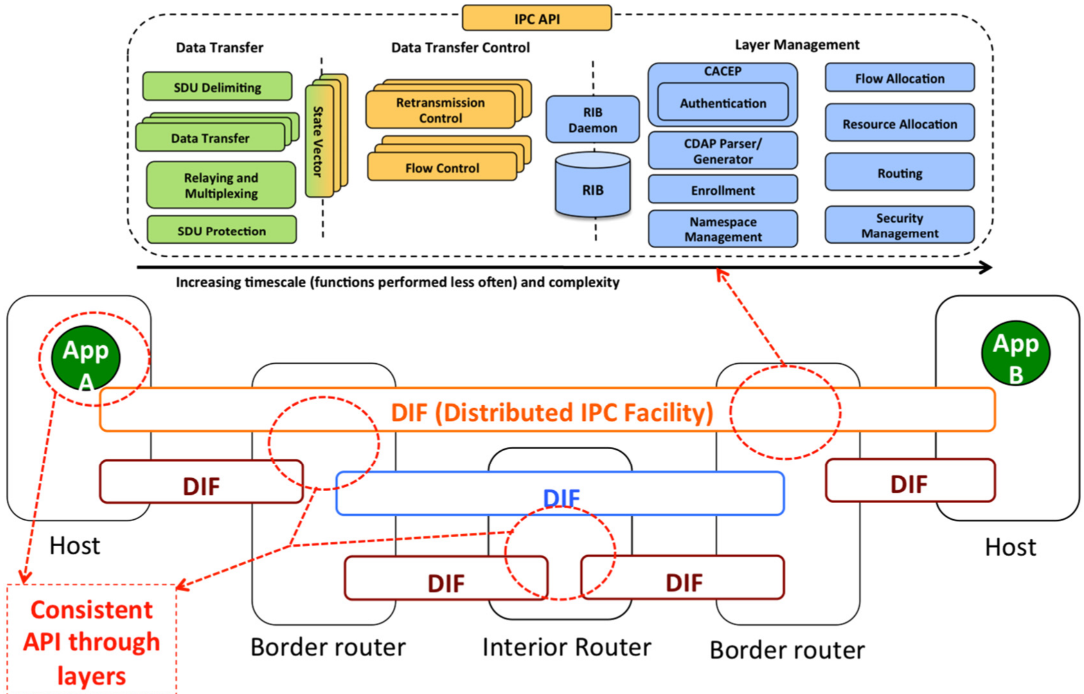

Fig. 1. Layer structure in RINA and functions of an IPC Process (IPCP).

remote management capabilities available via the Management Agent, IRATI provides a local text console to facilitate system administration of a local RINA instantiation.

The typical workflow of a RINA experiment using IRATI is the following. First the experimenter decides what is the design of the RINA network: how many nodes, how many DIFs (layers), which is the connectivity graph of each DIF, which are the policies of each DIF at each node and which applications will be deployed at each node. Once the design is ready, it needs to be translated to a configuration for each node. Next, assuming the IRATI software is already installed at each node and the proper configuration is in place, the software is instantiated, the different DIFs are put in place and experimentation can begin. The IRATI tutorials available at [27] provide detailed step-by-step guides of different experimental scenarios. Since carrying out these procedures manually is tedious and error-prone, a number of open-source tools to automate them have been developed; mainly the demonstrator [22] and Rumba [23,28]. ARCFIRE's deliverable 4.4 [29] report provides detailed information on how to use such tools to carry out large-scale RINA experiments using IRATI.

IRATI impact

In addition to the projects that have contributed to the development of IRATI (IRATI, PRISTINE, ARCFIRE), the ERASER [30] and OCARINA [31] projects are also using it to carry out their RINA-based research and development activities. The RINArmenia initiative [32] is evaluating the use of IRATI as a basis of the experimental part of their program. Moreover, IRATI has also been used by networking companies [17] to carry out Proof of Concept demonstrations of RINA in industry fora.

Moreover IRATI has also facilitated the development of other RINA implementations that have leveraged from the lessons learned by IRATI, such as: rlite [33] a lightweight RINA implementation that targets constrained environments such as IoT devices; or RINAsim [34], an OMNeT++ framework for simulating RINA networks.

Last but not least, IRATI has contributed to pushing the profile of RINA in standardization bodies, raising the interest of network operators in the technology. The improved RINA specifications to which IRATI development has heavily contributed are now under the procedure or ISO standardization within ISO SC6 WG7 [35–39]. Multiple RINA demonstrations using IRATI have been carried out as part of ETSI ISG Next Generation Protocols (NGP) group, crystallizing in a join ETSI publication with the European network operators Telefonica and Vodafone [2]. Ciena (a network equipment vendor) and BT (another network operator) carried out a public demonstration of RINA as a network slicint solution using IRATI at the Telecom Infra Project (TIP) Summit [17].

All in all has resulted in the following 12 scientific publications:

- The software architecture and main design choices of IRATI are explained in [40], while an initial performance evaluation is carried out in [41].
- A study that uses IRATI to demonstrate how RINA simplifies virtual machine networking is published in [26].
- [10] analyses how RINA reduces the cost of network management, analysing a datacentre configuration.
- The work in [7] illustrates the benefits of RINA for network security, showing how security policies can be designed implemented and deployed using IRATI.
- Programmability of RINA networks is explored in [9], describing the design of the IRATI's SDK and its application to develop congestion control and routing policies.
- RINA's flexibility for defining access control policies is reported in [12], backed up by experimental results using IRATI with a capability-based access control policy.
- The work in [6] demonstrates that network renumbering can be done in a fully automated fashion in RINA networks, without causing any impact to running flows.
- [5] experimentally verifies the RINA properties for mobility management, by carrying out a deployment of RINA over WiFi using IRATI.
- [14] reports about large scale experiments with IRATI to deliver differential Quality of Service to different types of applications, guaranteeing upper bounds on latency and delay on saturation.
- The work in [42] compares the overhead in joining the network and allocating a flow to an application for both RINA and TCP/IP, using IRATI for the experimental research.

- [43] describes the IRATI implementation of EFCP, the data transfer protocol framework of RINA.
- Finally, [15] documents a use case of RINA as a virtual networking solution, with a PoC of IRATI running as a Virtual Network Function on the equipment of a network vendor (Ciena).

The IRATI software has been instrumental in advancing the state of the art of RINA, the Recursive InterNetwork Architecture. The software project has fulfilled its goals of (i) becoming a platform for enhancing and completing the core RINA specifications; (ii) lowering the barriers of entry for researchers wanting to get involved in RINA R&D activities and (iii) enabling RINA PoCs with multiple industries. Moreover, IRATI has enlarged the community of individuals and organizations interested in RINA experimentation and deployment. IRATI has also served as a catalyst and source of experience for other RINA implementations, enriching the overall ecosystem.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Part of the design and development work of IRATI was funded by the European Commission through the H2020 ARCFIRE project (Grant 687871), the FP7 PRISTINE project (Grant 619305), the FP7 IRATI project (Grant 317814), and 2017 SGR 01658. The authors would like to acknowledge all the individuals and institutions that have also contributed to the design and implementation of IRATI: Miguel Ponce de Leon (Waterford Institute of Technology), Steve Bunch (TRIA Network Systems), Marc Suñé (Volta Networks), Kewin Rausch (FBK), Douwe de Bock (Ghent University), Julien Muchembled (VIFIB), Michael Crotty (Waterford Institute of Technology), Victor Álvarez (BISDN), Ondrej Lichtner (Brno University of Technology), Albert López (UPC), Berta Delgado (UPC), Marco Capitani (Nextworks) and Carolina Fernandez (i2CAT).

References

[1] J. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Commun. Mag. 49 (7) (2011) 26–36.
[2] ETSI Industry Specification Group on Next Generation Protocols (NGP) website. Available online at https://www.etsi.org/technologies-clusters/technologies/next-generation-protocols.
[3] J. Day, Patterns in Network Architecture, Prentice Hall, 2008.
[4] J. Day, I. Matta, K. Mattar, Networking is IPC: a guiding principle to a better internet, in: Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT '08, 2008.
[5] E. Grasa, L. Bergesio, M. Tarzan, D. Lopez, S. van der Meer, J. Day, L. Chitkushev, Mobility management in RINA networks: experimental validation of architectural properties, in: IEEE WCNC 2018, April 2018.
[6] E. Grasa, L. Bergesio, M. Tarzan, D. Lopez, J. Day, L. Chitkushev, Seamless network renumbering in RINA: automate address changes without breaking flows, in: European Conference on Networks and Communications, EUCNC, 2017.
[7] E. Grasa, O. Rysavy, O. Lichtner, H. Asgari, J. Day, L. Chitkushev, From protecting protocols to protecting layers: designing, implementing and experimenting with security policies in RINA, in: IEEE ICC 2016.
[8] J. Day, How in the Heck do you lose a layer? in: Network of the Future (NOF), 2011 International Conference on the, 28-30 Nov. 2011, pp. 135, 143.
[9] V. Maffione, F. Salvestrini, E. Grasa, L. Bergesio, M. Tarzan, A Software Development Kit to exploit RINA programmability, in: IEEE ICC 2016, Next Generation Networking and Internet Symposium.
[10] E. Grasa, B. Gaston, S. van der Meer, M. Crotty, M.A. Puente, Simplifying multilayer network management with RINA, TNC 2016, https://tnc16.geant.org/core/presentation/667.
[11] S. van der Meer, J. Keeney, L. Fallon, S. Feghhi, A. de Buitleir, 2019, Large-scale experimentation with network abstraction for network configuration management, in: RINA Workshop 2019, co-located with ICIN 2019 IEEEXplore.
[12] I. Amdouni, F. Hrizi, A. Laouiti, E. Grasa, H. Chaouchi, Exploring the flexibility of network access control in the recursive internetwork architecture, in: APCC 2016, Yogyakarta, 2016.
[13] Peyman Teymoori, Michael Welzl, Stein Gjessing, Eduard Grasa, Roberto Riggio, Kewin Rausch, Domenico Siracussa, Congestion control in the Recursive Internetwork Architecture (RINA). IEEE ICC 2016.
[14] J. Perelló, D. Careglio, A. López, Experimental with real application-specific QoS guarantees in a large-scale RINA demonstrator, in: 6th RINA International Workshop, co-located with ICIN 2019.
[15] M. Ponce de Leon, R. Ranganathan, D. Bainbridge, K. Ramanarayanan, A. Corston-Petrie, E. Grasa, Multi-operator IPC VPN slices: applying RINA to Overlay virtual networking, in: 6th RINA International Workshop, co-located with ICIN 2019.
[16] iporinad documentation: a daemon to allow the deployment of IP over RINA. Available online at: https://github.com/rlite/rlite#72-iporinad.
[17] Ciena corporation. Can orchestrating a network slice be as easy as 1-2-3? Available online at: https://www.ciena.com/insights/articles/Can-Orchestrating-an-End-to-End-E2E-Network-Slice-Be-as-Easy-as-1-2-3.html.
[18] IRATI github repository. Available online at: https://github.com/IRATI/stack.
[19] FP7 IRATI project website. Available online at: http://irati.eu/.
[20] FP7 PRISTINE project website. Available online at: http://ict-pristine.eu/.
[21] H2020 ARCFIRE project website. Available online at: http://ict-arcfire.eu/.
[22] Demonstrator tool repository. Available online at: https://github.com/IRATI/demonstrator.
[23] Rumba repository. Available online at: https://arcfire.gitlab.io/rumba/.
[24] RINA nginx – port of the popular webserver to the RINA API. Available online at: https://github.com/rlite/rina-nginx.
[25] RINA policies supported by IRATI and its configuration details. Available online at: https://github.com/IRATI/stack#322-dif-template-configuration-files.
[26] S. Vrijders, V. Maffione, D. Staessens, F. Salvestrini, M. Biancani, E. Grasa, D. Colle, M. Pickavet, J. Barron, J. Day, L. Chitkushev, Reducing the complexity of virtual machine networking, IEEE Commun. Mag. 54 (4) (2016) 152–158.
[27] IRATI tutorials. Available online at: https://github.com/IRATI/stack/wiki/Tutorials.
[28] S. Vrijders, D. Staessens, M. Capitano, V. Maffione, Rumba: a python framework for automating large-scale recursive internet experiments on GENI and FIRE+, in: IEEE Infocom Workshop on CNERT (Computer and Networking Experimental Research using Testbeds), 2018.
[29] ARCFIRE consortium. D4.4: Execution of experiments, analysis of results and bencharking against KPIs. Available online at: http://ict-arcfire.eu/index.php/research/deliverables/.
[30] ERASER project website. Available online at: http://people.ccaba.upc.edu/careglio/index.php/eraser-project/.
[31] OCARINA project website. Available online at: https://www.mn.uio.no/ifi/english/research/projects/ocarina/.
[32] RINArmenia initiative. Available online at: http://rinarmenia.com/.
[33] rlite github repository. Available online at: https://github.com/rlite/rlite.
[34] RINAsim, RINA simulator. Available online at: https://rinasim.omnetpp.org/.
[35] ISO/IEC CD 21558-1 Telecommunications and information exchange between systems – Future network architecture – Part 1: Reference model of recursive inter-network architecture. Available online at: https://www.iso.org/standard/78434.html?browse=tc.
[36] ISO/IEC CD 21559-1 Telecommunications and information exchange between systems – Future network protocols and mechanisms – Part 1: Recursive internetwork architecture error and flow control protocol. Available online at: https://www.iso.org/standard/78438.html?browse=tc.
[37] ISO/IEC CD 21559-2 Telecommunications and information exchange between systems – Future network protocols and mechanisms – Part 2: Recursive internetwork architecture flow allocator. Available online at: https://www.iso.org/standard/78438.html?browse=tc.
[38] ISO/IEC CD 21559-2 Telecommunications and information exchange between systems – Future network protocols and mechanisms – Part 3: Recursive internetwork architecture common application connection establishment. Available online at: https://www.iso.org/standard/78440.html?browse=tc.
[39] ISO/IEC CD 21559-2 Telecommunications and information exchange between systems – Future network protocols and mechanisms – Part 4: Recursive internetwork architecture common distributed application protocol. Available online at: https://www.iso.org/standard/78441.html?browse=tc.
[40] S. Vrijders, D. Staessens, D. Colle, F. Salvestrini, E. Grasa, M. Tarzan, L. Bergesio, Prototyping the recursive internet architecture: the IRATI project approach, IEEE Netw. 28 (2) (2014) 20,25.
[41] S. Vrijders, D. Staessens, D. Colle, F. Salvestrini, V. Maffione, L. Bergesio, M. Tarzan-Lorente, B. Gaston, E. Grasa, Experimental evaluation of a recursive internetwork architecture prototype, in: Global Communications Conference, GLOBECOM, IEEE, 2014.
[42] K. Ciko, M. Welzl, First contact: can switching to RINA save the Internet? in: 6th RINA International Workshop, co-located with ICIN 2019.
[43] M. Tarzan, E. Grasa, L. Bergesio, Error and flow control protocol design and implementation: A data transfer protocol for the recursive internet architecture, in: 6th RINA International Workshop, co-located with ICIN 2019.