An Invariant for Future Resilient Network Management Operations

Roland Bless
Karlsruhe Institute of Technology (KIT)

November 15, 2024

The complexity of today's networks is steadily increasing and so is the network management and control. The current efforts to achieve controllability by employing AI-based mechanisms and Network Digital Twins on top add even more complexity. This position paper argues to master the increasing complexity by simplification at the base: the introduction of the zero-touch control plane connectivity solution KIRA that serves as an invariant for network management operations. Its provided connectivity can be used as base for all management and control tasks allowing to recover from the most complex failure scenarios. KIRA's tightly integrated add-on services provide a perfect basis for autonomic network management solutions.

Keywords: Automation, orchestration, and autonomy

1 Introduction

Networks and networked services are getting more and more complex. Alongside this development the interdependencies for running network management, control, and operations are quite high today. For example, flow monitoring data is stored in some data bases and made available for management components that then will interact with the network. So a composition and chain of several services is used today to operate networks. If one of these services fail, this may cause others to fail as well [16] and complicate to analyze the root cause for the failure [24]. The situation is exacerbated by the increasing number of networked devices that need control and management. The current efforts to integrate AI-based mechanisms [12] and Network Digital Twins [30] on top add even more complexity. They will have multiple intertwined control loops at different levels that require lots of monitoring data to be processed.

Notable outages of large providers such as Meta [8], Amazon [27], Cloudflare [25], Rogers [28], or KDDI [15] indicate that even well-managed network infrastructures may experience failures that are hard to recover from due to the complexity of controlling the network infrastructure and intricate interdependencies [16]. Especially, the outages at Meta, Cloudflare, and Rogers were caused by (mostly BGP) configuration errors that prevented network operators from accessing their control network, thereby inhibiting them to diagnose the root cause and fix the failure. Regaining control over the network is then a tedious process, potentially requiring direct on-site access to the devices in order to revert the erroneous changes.

This clearly motivates the need for resilient control and management solutions that avoid manual configuration and that are able to adapt to changing conditions and to self-heal. Several approaches such as Autonomic Networking and Management [1, 20], Zero Touch Management [4], and Self-Driving Management [6] strive for more *autonomic* (i.e., self-managing [1]) solutions. However, they assume that control plane connectivity is available.

Basically, there are two different ways to provide control plane connectivity. Large and complex networks make it nearly impractical to use separate *out-of-band control plane networks (CPNs)* that need their own devices, setup, and configuration. Out-of-band CPN need to be "highly available, easy to manage and maintain, and cost effective" [10], but come with the burden of installing and operating two distinct networks. An *in-band* CPN uses the same links for control as for transporting the data packets and is cheaper, but comes at the cost of potential circular dependencies on connectivity [5]. Both CPN variants need a connectivity solution that often requires a routing solution for larger networks (smaller networks may simply use a link-layer solution). [10] and [5] use a hybrid CPN approach, i.e., a mixture of in-band and out-of-band CPNs. [5] uses Open/R [9] as fallback connectivity solution. In-band CPNs nevertheless require prioritizing traffic of the routing protocol that provides the connectivity.

This position paper proposes to introduce a resilient invariant for control and network management that has no configuration dependencies (zero-touch) and provides control plane connectivity unconditionally[1]. When this solution is deployed, it would make network management and control more robust, because it always offers the possibility to reset devices, their configurations, and services to a well-known state in case of failures. It therefore serves as *connectivity invariant* to bootstrap networked resources and services as well as to recover from failures. Today's and future networks are possibly too complex to preclude configuration mistakes and resulting failures. Tools to audit configuration changes are used, but have shown to fail in practice [8, 25], too. It is probably time to start a transition to resilient network management and control that allows failures to happen, but that is able to recover from them realiably by using the control plane connectivity invariant that is only self-dependent.

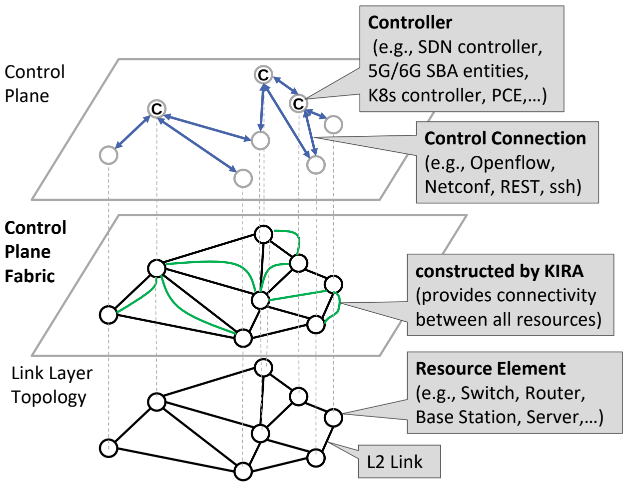

Figure 1: KIRA constructs a Control Plane Fabric in a Zero-touch Manner

2 KIRA as Invariant for Control Plane Connectivity

KIRA [3, 2] is a scalable zero-touch routing architecture that provides IPv6 connectivity without any manual configuration across all different kinds of topologies. Zero-touch does not only mean without manual configuration, but also includes adaptivity. In this context KIRA adapts automatically to different underlay topologies and link or node failures in a self-organizing manner. It is ID-based, i.e., network resources keep their address even while changing their connectivity in the topology, e.g., by moving across the topology or becoming multi-homed. It provides self-generated addresses (currently using a 16-bit ULA [14] prefix and a 112 bit NodeID that is randomly generated), therefore it does not need any other address assignment mechanism for building its connectivity.

It builds a *control plane fabric* (see also [16]) on top an underlying (usually link layer) topology as illustrated in fig. 1. Control and management entities can exert control over their resources on top of this connectivity, e.g., by creating control connections and sending commands to the resources or gathering monitoring data and so on.

Design and Features

KIRA consists of a two tier architecture shown in fig. 2. The *Routing Tier* consists of the ID-based routing protocol R^2/Kad that is based on the Kademlia peer-to-peer overlay approach. It uses XOR as distance metric between NodeIDs, path vectors, and source routing. Its routing tables are only growing with $\mathcal{O}(\log(n))$, where n is the number of nodes in the network. Therefore it scales very well, but as trade-off routes may incur some stretch (i.e., they are longer compared to the shortest path). However, the average stretch is acceptable for general control plane traffic. Moreover, a route to a contact in the routing table of a node is converging to a shortest path route. Therefore, KIRA prioritizes connectivity over route efficiency. KIRA includes a scheme for path rediscovery in case links or nodes fail. More details of how R^2/Kad works can be found in [3].

Source routing is very robust, but may imply high per packet overhead. Therefore, KIRA uses a *Forwarding Tier* for efficiently forwarding data packets of the CPN, i.e., the actual control and management traffic. PathIDs are used instead of source routes and consist of a hash of the NodeIDs along the complete path. PathIDs are precomputed in a 2-hop vicinity, therefore only for paths longer than 5 hops PathIDs must be installed in intermediate systems in order to swap the PathIDs similar to label swapping. KIRA's Forwarding Tier can simply use existing forwarding plane technology that is IPv6 capable and can also use either GRE [21] or SRv6 [11] for encapsulation. As indicated in fig. 2, control and management applications can simply use the IPv6 connectivity provided by KIRA, i.e., they can use any transport protocol that works with IPv6, so they do not need to be adapted to work with KIRA.

KIRA has some interesting properties that make it very well suited as connectivity solution for control plane fabrics:

- KIRA is loop-free in the sense that packets will never cycle in an "endless" loop (as with hop-by-hop routing that employs hop limit mechanisms

[1] the underlying connectivity must be working to some extent though

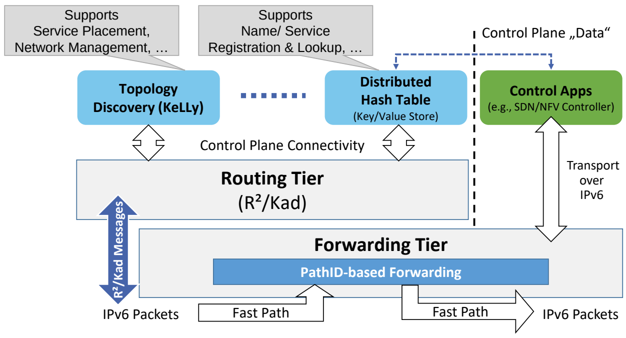

Figure 2: Kira's Two Tier Architecture

as mitigation). Moreover, it is loop-free even during convergence.

- It enables a *per-node* decision on *stretch/memory trade-off* by allowing to put additional entries into its routing table, e.g., to get more efficient routes to resources that need to be controlled by a control application running on a Kira node.
- It is multi-path capable due to small routing tables and expressiveness of the source routes.
- It can support fast reroute without loops due to source routes.
- It supports different routing metrics.
- It has a built-in route flapping prevention in the sense that it will not alternate between two equally good routes.
- It includes a specific end-system mode for non-routing nodes.
- It supports *Domains* so that its confines routes to and via nodes within the corresponding domain. Domains may be defined by topological or organizational criteria. However, they have to be assigned to nodes. The NodeID stays the same in all domains and there exists always a global domain that can be used for further configuration. Domains may also be statically assigned together with the software image or by the geographical location.

Kira also supports additional services that are useful for control and management. They are not part of Kira itself, but run as tightly coupled modules as shown in 2. First, it supports a *Distributed Hash Table (DHT)* that can be used to store (key,value) pairs, e.g., to provide a simple name service that maps human readable names to NodeIDs (that are randomly generated). The DHT can also help to register and find service instances, i.e., it enables service discovery. Second, Kira provides a very efficient topology discovery mechanism called KeLLy [23] that can be used for controller placement, service orchestration, or creating areas and so on. Topology discovery is an essential part of modern architectures [5, 16]. These supporting and tightly integrated add-on services make it easily possible to bootstrap management and control entities, letting them rendezvous and self-organize for executing truly distributed control.

Existing Solutions

To best of our knowledge, there do not exist comparable scalable zero-touch routing protocol solutions that provide similar features like Kira. The ANIMA working group uses RPL [29] in a specific profile [19] as connectivity solution for the Autonomic Control Plane [7]. RPL, however, constructs a tree-like routing structure and requires manual assignment and configuration of root nodes. Because many routes lead via the root node, RPL produces heavy traffic concentration near the root node as well as stretch [26] and the routing table size at and near the root node grows with $\mathcal{O}(n)$. More efficient routes (e.g., using the point-ot-point route discovery mode [13]) come at the cost of additional entries and also additional route discovery overhead.

Existing autoconfiguration solutions for OSPF [17] or IS-IS [18] are intended for use in small deployments only (10s of devices) and do not work for large deployments that require division into multiple areas. Self-organization of the latter is complex [22] and not yet solved. Open/R [9] is supposed to scale to 10 000 nodes, but requires to introduce areas beyond this size. Discussion of more related work can be found in [3].

3 Conclusions

Using Kira as invariant for control plane connectivity can be seen as enabler for much more robust network management operations. Network operators do not need to know how their current network looks like in advance as Kira is able to provide up-to-date topology information. Its provided connectivity can be used as base for all management and control tasks, its tightly integrated add-on services provide a perfect basis for autonomic network management solutions. Network operators should never loose control over their network infrastructure and be able to recover from even the most complex failure scenarios. A next step toward realizing this vision would be the standardization of Kira by the IETF so that it is available in all networked devices that need to be managed or controlled.

References

[1] M. Behringer, M. Pritikin, S. Bjarnason, A. Clemm, B. Carpenter, S. Jiang, and L. Ciavaglia. Autonomic Networking: Definitions and Design Goals. RFC 7575 (Informational), June 2015.

[2] R. Bless. KIRA Project Web Page. `https://s.kit.edu/KIRA`, nov 2024.

[3] R. Bless, M. Zitterbart, Z. Despotovic, and A. Hecker. KIRA: Distributed Scalable ID-based Routing with Fast Forwarding. In *2022 IFIP Networking Conference (IFIP Networking)*, pages 1–9, Catania, Italy, June 2022.

[4] E. Coronado, R. Behravesh, T. Subramanya, A. Fernàndez-Fernàndez, M. S. Siddiqui, X. Costa-Pérez, and R. Riggio. Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks. *IEEE Communications Surveys & Tutorials*, 24(4):2535–2578, 2022.

[5] M. Denis et al. EBB: Reliable and Evolvable Express Backbone Network in Meta. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, page 346–359, New York, NY, USA, 2023. Association for Computing Machinery.

[6] K. Dzeparoska, N. Beigi-Mohammadi, A. Tizghadam, and A. Leon-Garcia. Towards a Self-Driving Management System for the Automated Realization of Intents. *IEEE Access*, 9:159882–159907, 2021.

[7] T. Eckert (Ed.), M. Behringer (Ed.), and S. Bjarnason. An Autonomic Control Plane (ACP). RFC 8994 (Proposed Standard), May 2021.

[8] Engineering at Meta. More details about the October 4 outage. `https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/`, Oct 2021.

[9] FaceBook. Open Routing – Open/R Documentation. `https://openr.readthedocs.io/`, Dec. 2023.

[10] A. D. Ferguson et al. Orion: Google's Software-Defined Networking Control Plane. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 83–98. USENIX Association, Apr. 2021.

[11] C. Filsfils (Ed.), D. Dukes (Ed.), S. Previdi, J. Leddy, S. Matsushima, and D. Voyer. IPv6 Segment Routing Header (SRH). RFC 8754 (Proposed Standard), Mar. 2020.

[12] J. François, A. Clemm, D. Papadimitriou, S. Fernandes, and S. Schneider. Research Challenges in Coupling Artificial Intelligence and Network Management. Internet-Draft draft-irtf-nmrg-ai-challenges-03, Internet Engineering Task Force, Mar. 2024. Work in Progress.

[13] M. Goyal (Ed.), E. Baccelli, M. Philipp, A. Brandt, and J. Martocci. Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks. RFC 6997 (Experimental), Aug. 2013.

[14] R. Hinden and B. Haberman. Unique Local IPv6 Unicast Addresses. RFC 4193 (Proposed Standard), Oct. 2005.

[15] KDDI Corporation. The July 2 Communication Failure and Our Response. `https://www.kddi.com/english/important-news/20220729_01/`, July 2022.

[16] A. Krentsel, N. Saran, B. Koley, S. Mandal, A. Narayanan, S. Ratnasamy, A. Al-Shabibi, A. Shaikh, R. Shakir, A. Singla, and H. Weatherspoon. A decentralized sdn architecture for the wan. In *Proceedings of the ACM SIGCOMM 2024 Conference*, ACM SIGCOMM '24, page 938–953, New York, NY, USA, 2024. Association for Computing Machinery.

[17] A. Lindem and J. Arkko. OSPFv3 Autoconfiguration. RFC 7503 (Proposed Standard), Apr. 2015.

[18] B. Liu (Ed.), L. Ginsberg, B. Decraene, I. Farrer, and M. Abrahamsson. IS-IS Autoconfiguration. RFC 8196 (Proposed Standard), July 2017.

[19] H. Mahrt and M. Zitterbart. RPL-based Control Plane Connectivity: An Evaluation. In *Proceedings of 14th International Conference on Network of the Future (NoF)*. IEEE, Nov. 2023.

[20] T. B. Meriem, R. Chaparadza, B. Radier, S. Soulhi, J.-A. Lozano-López, and A. Prakash. GANA – Generic Autonomic Networking Architecture. ETSI Whitepaper No. 16, Oct. 2016. ISBN 979-10-92620-10-8.

[21] C. Pignataro, R. Bonica, and S. Krishnan. IPv6 Support for Generic Routing Encapsulation (GRE). RFC 7676 (Proposed Standard), Oct. 2015.

[22] G. Rétvári, F. Németh, R. Chaparadza, and R. Szabó. Ospf for implementing self-adaptive routing in autonomic networks: A case study. In J. C. Strassner and Y. M. Ghamri-Doudane, editors, *Modelling Autonomic Communications Environments*, pages 72–85. Springer Berlin Heidelberg, 2009.

[23] P. Seehofer, R. Bless, H. Mahrt, and M. Zitterbart. Scalable and Efficient Link Layer Topology Discovery for Autonomic Networks. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–9, Nov. 2023.

[24] A. W. Services. Summary of the AWS Service Event in the Northern Virginia (US-EAST-1) Region. https://aws.amazon.com/de/message/12721/, Dec. 2021.

[25] T. Strickx and J. Hartman. Cloudflare outage on June 21, 2022. The Cloudflare Blog https://blog.cloudflare.com/cloudflare-outage-on-june-21-2022/, June 2022.

[26] J. Tripathi (Ed.), J. de Oliveira (Ed.), and J. Vasseur (Ed.). Performance Evaluation of the Routing Protocol for Low-Power and Lossy Networks (RPL). RFC 6687 (Informational), Oct. 2012.

[27] Washington Post. Amazon Web Services' third outage in a month exposes a weak point in the Internet's backbone. https://www.washingtonpost.com/business/2021/12/22/amazon-web-services-experiences-another-big-outage/, Dec. 2021.

[28] Wikipedia. 2022 Rogers Communications outage. https://en.wikipedia.org/wiki/2022_Rogers_Communications_outage, Sept. 2022.

[29] T. Winter (Ed.), P. Thubert (Ed.), A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550 (Proposed Standard), Mar. 2012. Updated by RFCs 9008, 9010.

[30] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet. Network Digital Twin: Concepts and Reference Architecture. Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-08, Internet Engineering Task Force, Oct. 2024. Work in Progress.