Toward Content-Oriented Orchestration: SDN and NFV as Enabling Technologies for NDN

Hoang Long Mai*, Messaoud Aouadj*, Guillaume Doyen*, Wissam Mallouli†, Edgardo Montes de Oca†, Olivier Festor‡
*ICD - CNRS UMR 6281, Troyes University of Technology, 10004 Troyes - France
†Montimage, 39 rue Bobillot, 75013 Paris - France
‡LORIA - CNRS UMR 7503, TELECOM Nancy - University of Lorraine, 54506 Vandoeuvre-les-Nancy - France

***Abstract*—Network Function Virtualization (NFV) is a novel paradigm which enables the deployment of network functions on commodity hardware. As such, it also stands for a deployment enabler for any novel networking function or networking paradigm such as Named Data Networking (NDN), the most promising solution relying on the Information-Centric Networking (ICN) paradigm. However, dedicated solutions for the security and performance orchestration of such an emerging paradigm are still lacking thus preventing its adoption by network operators. In this paper, we propose a first step toward a content-oriented orchestration whose purpose is to deploy, manage and secure an NDN virtual network. We present the way we leverage the TOSCA standard, using a crafted NDN oriented extension to enable the specification of both deployment and operational behavior requirements of NDN services. We also highlight NDN-related security and performance policies to produce counter-measures against anomalies that can either come from attacks or performance incidents.**

***Index Terms*—SDN, NFV, NDN, Orchestration, MANO, Performance, Security.**

I. Introduction

Named Data Networking (NDN) is a networking paradigm proposing an Internet data plane that shifts from host-based network mechanisms to content-based ones. We argue that the Network Function Virtualization (NFV) model, allowing network operators to deploy network functions in software that runs on a standard commodity server hardware, coupled with SDN is a good candidate to support ICN deployment.

Although the main Management and Orchestration (MANO) functional blocks are now standardized by ETSI work and implemented by current NFV frameworks, they primarily target IP-based Virtual Network Functions (VNF). Consequently, hosting non-IP VNF such as NDN ones impacts the management plane and functions, requiring MANO extensions. We present a novel NFV MANO architecture with orchestration components dedicated to the deployment, management, monitoring, and security of an NDN virtual network. The main building blocks of our solution rely on an extended TOSCA profile for NDN which also integrates novel policies to dynamically react against NDN specific performance and security incidents.

To present and assess our solution, the paper is organized as follows. First, we present in Section II the necessary background on ICN/NDN and NFV. Next, in Section III the architecture of our NFV MANO framework as well as its components are depicted. Afterward, in Section IV, we present the different results obtained from different experiments we performed to evaluate the performance of our content-oriented orchestration. Finally, we draw a set of conclusions and outline future work in Section V.

978-3-903176-15-7 © 2019 IFIP

II. Background and Related works

A. Information Centric Networking

Information Centric Networking [1] (ICN) is a networking architecture performing a shift from the traditional host-to-host communication paradigm of the Internet to a host-to-content one. Among all ICN approaches, NDN (Named Data Networking) [2] is the most mature and adopted one. The NDN architecture employs a pull-based mechanism, using two kinds of packets, *Interest* and *Data*, which stand for request and response packets respectively. An NDN node consists of three components: a *Forwarding Information Base (FIB)*, a *Pending Interest Table (PIT)*, and a *Content Store (CS)*. The *FIB* maintains routing information to forward *Interests*. The *PIT* keeps all pending *Interests* and their arrival faces into entries, and each entry is removed when the matching *Data* is received or when a timeout occurs. The *CS* is used for caching *Data* packets with a caching replacement policy.

Although NDN was designed to tackle security issues inherent to IP networks, it also induces several new security attacks. Among them, the *Content Poisoning Attack (CPA)* [3], [4] has been identified as a major threat by the NDN community. In *CPA*, a malicious producer with the help of a malicious consumer force any NDN node on their path to insert altered content in its *CS*, making it answering to its downstream nodes with this altered content, thus making the pollution of *CS* in NDN nodes possible.

B. ICN Deployment and cohabitation with IP

The different proposals for the cohabitation of IP and ICN offered by SDN differ from the perspective of: (1) the data-plane packet encapsulation of ICN traffic (overlay vs. native) (2) the type of forwarding tables hosted by forwarding elements (IP flow table, *ICN FIB* or both), (3) the awareness of ICN from SDN controllers, and (4) the control protocol used to carry control information between forwarding elements and the SDN controller (OpenFlow vs. a dedicated one). In [5], the authors propose to use a dedicated UDP or TCP port to identify the ICN protocol and to extend the SDN controller with an ICN module. Salsano et al. [6] propose a framework for deploying ICN functionalities over SDN using the IP option header to server as the name field for ICN. Meanwhile, the authors of [7] propose and implement an ICN module in the SDN controller to process the forwarding path computation for NDN flows, separately from the IP flows. Nguyen et al. [8] implement an intermediate layer between a CCN node and an OpenFlow switch. The combination of these three elements acts as an ICN router.

NFV rather argues for the separation of the IP and ICN protocols by leveraging the isolation property of virtualization [9]. Sardara et al. [10] follow this direction by proposing vICN (Virtualized ICN)[1]. The authors provide a flexible unified framework for ICN, which includes several functions such as monitoring. The 5G next-generation architecture [11] leverages NFV and SDN technologies to provide the flexibility to deploy ICN-as-a-Slice. Finally, in [12], the authors take benefits of NFV to provide contextualized edge services relying on ICN protocol stacks. However, to the best of our knowledge, our architecture [13], [14] is the first approach which pushes the content-oriented paradigm of ICN up to high-level orchestration templates.

III. Management and Orchestration for NDN

The overall architecture we implemented strictly follows the ETSI reference architecture specification [15] and considers Docker, a well-adopted and efficient solution, as the core technology for the *Network Functions Virtualization Infrastructure (NFVI)*. As an encapsulation strategy for the NDN data-plane traffic, it considers VXLAN, thus making the *NFVI* agnostic to the carried traffic nature and thus enabling it to carry both NDN and IP traffic. Consequently, the *Virtual Infrastructure Manager (VIM)* in charge of controlling and managing the *NFVI* compute, storage, and network resources, does not need to be extended to support NDN traffic. Hence, we have selected Docker Swarm as a ready-to-use technology.

A. NDN TOSCA profile and policies

Our TOSCA profile extends the OASIS simple profile for NFV to take into account NDN features, thus providing some base types for modeling an end-to-end NDN service deployment and operational behavior. Basically, a TOSCA specification provides the following base nodes and policies to construct a service topology: *Virtual Deployment Unit (VDU), VNF, Virtual Link (VL), Connection Point (CP)*, *Forwarding Path* and policies. In our case, we reuse the three following standard nodes without any modification: *VDU, VL* and *CP* to model a virtualized NDN network, as they relate only to the infrastructure layer which is NDN agnostic. Rather, to model the forwarding elements of a virtualized NDN network, the TOSCA nodes for *VNF*, *Forwarding Path* and TOSCA policies have been extended.

Our *VNF* node includes configuration parameters that represent the set of NDN prefixes to be announced as well as the status of a signature verification module in a NDN router. We have also included additional information in the *VNF* node to consider specific properties of particular NDN components. Secondly, the *Forwarding Path* specification has also been extended to capture the list of VNFs that a particular set of NDN packets will follow.

Our NDN TOSCA extension enables the specification policies modeled with Event-Condition-Action (ECA) rules that apply dynamically during service runtime. The NDN TOSCA profile includes two types of policies which deal with dynamic re-configuration operations and scale-out. The configuration policy allows to dynamically change the state of an NDN node. Each policy includes the identifiers of the target VNF as policy enforcement points and the event which triggers this action, which is, in the case of security policies, an alert issued by a detection engine. As a second example of a dynamic reconfiguration policy, one can consider the update of the white and blacklist of an NDN firewall. Finally, we have defined a scale-out policy which allows any NDN routing component to be dynamically replicated as soon as the PIT size of a given NDN router crosses a given threshold. This generates a dedicated event which triggers the countermeasure.

B. Content aware components

The methodology we followed has consisted in solely extending or redesigning the MANO components which need NDN awareness, without diverting them from their initial purpose. These are the *VNF Manager (VNFM)* and the *NFV Orchestrator (NFVO)*. They are extensively described in the context of the deployment and management of an NDN virtual network.

1) NDN VNFs: In our architecture, VNFs are dockerized applications, which include an NDN router and an NDN firewall. A monitoring probe has been developed and integrated into each *NDN VNF* to collect and correlate different NDN metrics to identify anomalies or potential attacks [16]. Besides, another security middle-box dedicated to signature verification is proposed in [14] has also been integrated. Consequently, we considered a standalone NDN firewall as a VNF whose presentation can be found in [17]. This firewall can be easily configured by our orchestration solution to append or delete filtering rules.

2) NFVO: In the context of NDN, the *NFVO* main functionalities must be rethought to take into consideration the features of this new paradigm. To integrate them, we have designed and implemented a dedicated *NFVO* which reads TOSCA templates and creates an in-memory graph of TOSCA nodes and their relationships and especially the NDN configuration in an NDN TOSCA profile. Beside the initial configuration defined in the deployment specification of an NDN TOSCA profile, the *NFVO* also considers actions to perform from NDN's TOSCA policies.

3) VNF Configuration Management: In the context of NDN, the *VNFM* is extended to forward all specific NDN

[1] https://wiki.fd.io/view/Vicn

configurations from the *NFVO* to *NDN VNFs*. It is also extended to receive notifications from the different *Element Managers (EM)*, which are the internal components responsible for all management purposes in a *NDN VNF*. In addition, the *EM* also includes a local controller unit (*Local Controller*) which represents a local point of control and embeds security applications that allow it to detect NDN attacks locally.

C. Content-Oriented Orchestration

Finally, we now explain the two main orchestration processes that are the deployment automation of an NDN virtual network and the dynamic enforcement of policies, leading to reconfiguration operations for performance or security purposes.

1) Deployment automation and dynamic configuration of functions: In the deployment phase, firstly, the *VIM* deploys a management network to ensure the communication between VNFs and the *VNFM*, followed by the deployment of *VNFM*'s container connecting to this network. Next, the *VNFM* updates the routes for prefixes in the *FIB* thanks to the *EM*. One issue that can occur at this stage is that the connection between VNFs may not be ready when the router finishes to receives its NDN configuration from the *VNFM*. To solve this issue, periodically, each *EM* verifies the *FIB* of its related NDN router to ensure that the *NFD* process already takes into account the configuration received from the *VNFM* and it proceeds to a reconfiguration if needed. Triggered by an event defined in the TOSCA policies, the *NFVO* can also dynamically reconfigure any virtualized NDN function with a process which is similar to the deployment.

2) Scale-out: Scaling-out a service is an essential feature for a MANO architecture to guarantee the performance and network throughput. To ease the understanding, we illustrate it in the context of a router *SV-Router* which performs signature verification and whose load becomes consequently high. *SV-Router* periodically sends the number of entities in its *PIT* to the *VNFM* which in turn forwards it to the *NFVO*. Whenever a given threshold is crossed, the *NFVO* commands *VIM* and *NFVI* to scale-out the *VNF containers* to a specified number of replicas which is defined in the TOSCA profile. Finally, the forwarding strategy in this node is changed to *round-robin* which allows balancing the load into the different replicas of the *SV-Router*. As the scale-out process is performed progressively, the network continues to provide the service without any disruption.

IV. Evaluation

In this section, we present the results we have collected in different experiments, which have been conducted in a virtual testbed hosted over OpenStack in which each Point of Presence (PoP) is emulated by a virtual machine hosting an NFVI and among them, one is selected to host our content-oriented orchestrator which dynamically spawns the VNFM as a Docker container. We evaluate the capability of our content orchestrated solution to dynamically react to events and its capacity to dynamically reconfigure the virtual network and its VNFs accordingly. One can note that all presented results are the average of five repetitions bounded by 95% confidence intervals.

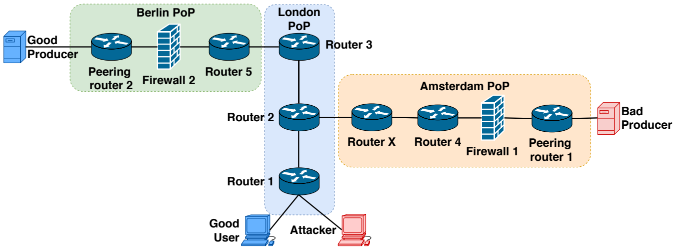

Fig. 1. Experiment topology

A. Use case

The topology we consider, depicted in Figure 1, is an extract of the ClaraNet operator topology provided by the Internet Zoo Topology dataset [18]. Here, we consider two different VNFs presented in Section III.B.1 which are NDN routers and NDN firewalls. Our three PoPs respectively host edge access nodes (in Berlin and Amsterdam), containing a peering node, a firewall and a router, and a core network (in London) containing three core NDN routers. A good producer of NDN content is connected to the Berlin PoP while a malicious one is connected to Amsterdam. The London PoP connects both a good user and an attacker whose purpose consists in corrupting the data hosted in all network caches to prevent the good user to access any desired content (CPA).

B. Orchestration scenarios

In order to evaluate the dynamic part of our orchestration framework, we consider the case of a content poisoning attack. The detection of this attack is achieved through a dedicated detector presented in [19] and from an orchestration perspective, we consider three mitigation scenarios: *No policy*, *No scale-out* and *Scale-out*. In the case of *No policy*, the entire network is deployed without any policy to mitigate security threats. This scenario is considered as a reference to evaluate the relevance of the subsequent ones. In the second scenario (*No scale-out*), the *NFVO* enables a policy that allows the dynamic configuration of the firewall and the dynamic activation of signature verification in NDN routers. Finally, in the *Scale-out* scenario, the *NFVO* enforces the mitigation policies presented above and it also enforces a scale-out operation on NDN routers which suffers from a too high computing load.

The constant parameters that were used to run these three scenarios are reused from experiments in [3] and the following scripts illustrate the policies that were used in the case of the *Scale-out* one. The signature verification policy is the one triggered by *Router 2* to activate the verification signature in targets *Router 4* and *Router 5* which are the best candidate to enforce a signature verification that can prevent corrupted content to be pushed by the bad producer. Then, the firewall policy is the triggered by *Router 4* and *Router 5* and it allows the configuration of *Firewall 1* and *Firewall 2* to be dynamically augmented with a rule to block the corrupted

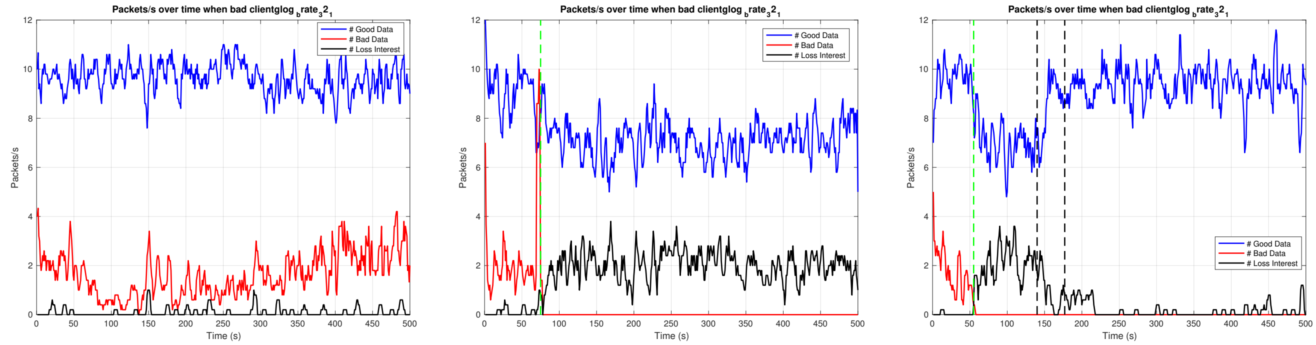

Fig. 2. Snapshot of Good/Bad/Lost data in scenarios No Policy - No Scale-out and Scale-out

content. Finally, the following policy provides an example of performance policy which allows scaling out the VNFs *Router 4 Router 5* to three replicas whenever a given threshold is crossed, which is the number of entries in the PIT of NDN routers.

```
scaling_out_policy:
  type: tosca.policies.nfv.doctor.ndn.scaling
  targets: [router_4, router_5]
  triggers:
   scale_out:
      meter_name: PIT
      event_type: tosca.policies.nfv.doctor.ndn.n_pit
      condition:
       constraint: pending_interests greater_than 5
       threshold: 5
       comparison_operator: gt
       period: 10
      action:
       action_type: scale_out
       number: 3
```

1) Numerical results: Figure 2 exhibits time series of *Good Data* (blue lines), *Bad Data* (red lines) and *Lost Data* (black lines), expressed in terms of number of packets per second, received by the good user in the context of the three scenarios described above.

In the *No policy* scenario, the frequency for *Good Data* is approximately 10 packets/s. However, there is also a substantial amount of *Bad Data*, about 3 packets/s. This scenario demonstrates that without mitigation, the attack can corrupt up to one-third of the content provided to a good user.

In the *No scale-out* scenario, the green line shows the time at which the network detects the attack and triggers the firewall and signature verification policies to mitigate the attack. We can observe that after this moment, the number of *Bad Data* abruptly drops to zero. However, the number of *Good Data* also decreases slightly, and the number of *Lost Data* increases remarkably. The reason for this phenomenon lies in the overload the routers performing the signature verification suffers from, which prevent it from reliably achieve its basic forwarding operations.

In the last scenario, *Scale-out*, the two dark lines show the instant at which the scale-out configuration operations start and terminate. In this scenario, one notices that the mitigation efficiency is identical to the *No scale-out* scenario, with excellent filtering of corrupted data. Moreover, when the scale-out is enforced, the network returns to a normal state: *Good Data* gets back to 10 packets/s, and *Lost Data* drops to a negligible value. We can also observe that during the period of scale-out, there is no downtime of the network.

Beyond these illustrations of mitigation, over all our experiments, the percentage of *Bad Data* in the *No Scale-out* and *Scale-out* scenarios are smaller than in *No Policy* which demonstrates the efficiency of both the signature verification and the firewall reconfiguration policies. The percentage of *Bad Data* which remains on these two scenarios is the percentage of *Bad Data* before the attack detection. After the activation of these policies, the number of *Bad Data* drops to zero. On the other hand, in the *Scale-out* scenario, the percentage of *Bad Data* and *Lost Data* after the scale-out operation decrease, which proves the relevance of the scale-out policy.

V. Conclusion and Future Work

Deploying NDN is a strategic and complex task which can be supported by a combination of SDN/NFV and smart orchestration. In this paper, we have defined the core elements for the design and implementation of a content-based orchestration for virtualized NDN networks. We have shown that the TOSCA standard can be extended to become a solid content-based service specification template. With novel orchestration components, we have demonstrated that a content-oriented orchestrator can both automatically deploy a virtual NDN topology, but also enforce dynamic reconfiguration in the virtual network triggered by security and scale-out policies. We have assessed the architecture on a real deployment and collected both functional proof and measurements of the viability of the proposed system.

Our future work will focus on further developing content oriented service specifications that can extend the control of data-plane operations in the virtual network over features that go beyond name prefixes (e.g., content type) to orchestrate new and specific routing and caching policies.

Acknowledgment

This work is partially co-funded by (1) the French National Research Agency (ANR), DOCTOR project, <ANR-14- CE28-0001>, started in 01/12/2014 and supported by the French Systematic cluster, (2) the CRCA Excellence program <A2101-03>, and (3) the CRCA and FEDER CyberSec Platform, <201304601>.

REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, July 2012.

[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[3] T. Nguyen, X. Marchal, G. Doyen, T. Cholez, and R. Cogranne, "Content poisoning in named data networking: Comprehensive characterization of real deployment," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 72–80.

[4] Z. Xu, B. Chen, N. Wang, Y. Zhang, and Z. Li, "Elda: Towards efficient and lightweight detection of cache pollution attacks in ndn," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*. IEEE, 2015, pp. 82–90.

[5] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling information centric networking in ip networks using sdn," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–6.

[6] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, "Information centric networking over sdn and openflow: Architectural aspects and experiments on the ofelia testbed," *Computer Networks*, vol. 57, no. 16, pp. 3207 – 3221, 2013, information Centric Networking. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613002727

[7] N. L. M. van Adrichem and F. A. Kuipers, "Ndnflow: Software-defined named data networking," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–5.

[8] X. N. Nguyen, D. Saucez, and T. Turletti, "Efficient caching in content-centric networks using openflow," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 1–2.

[9] A. Rahman, D. Trossen, D. Kutscher, and R. Ravindran, "Deployment considerations for information-centric networking (icn)," *ICNRG draft, available at https://tools.ietf.org/html/draft-irtf-icnrg-deployment-guidelines-04*, 2018.

[10] M. Sardara, L. Muscariello, J. Augé, M. Enguehard, A. Compagno, and G. Carofiglio, "Virtualized icn (vicn): Towards a unified network virtualization framework for icn experimentation," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ser. ICN '17. New York, NY, USA: ACM, 2017, pp. 109–115. [Online]. Available: http://doi.acm.org/10.1145/3125719.3125726

[11] 3GPP, """technical specification group services and system aspects; system architecture for the 5g system (rel.15)," 2017.

[12] P. TalebiFard, R. Ravindran, A. Chakraborti, J. Pan, A. Mercian, G. Wang, and V. C. M. Leung, "An information centric networking approach towards contextualized edge service," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 250–255.

[13] B. Mathieu, G. Doyen, W. Mallouli, T. Silverston, O. Bettan, F. X. Aguessy, T. Cholez, A. Lahmadi, P. Truong, and E. M. d. Oca, "Monitoring and securing new functions deployed in a virtualized networking environment," in *2015 10th International Conference on Availability, Reliability and Security*, Aug 2015, pp. 741–748.

[14] X. Marchal, M. E. Aoun, B. Mathieu, W. Mallouli, T. Cholez, G. Doyen, P. Truong, A. Ploix, and E. M. De Oca, "A virtualized and monitored ndn infrastructure featuring a ndn/http gateway," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '16. New York, NY, USA: ACM, 2016, pp. 225–226. [Online]. Available: http://doi.acm.org/10.1145/2984356.2985238

[15] European telecommunications standards institute, industry specification groups nfv. [Online]. Available: https://www.etsi.org/technologies-clusters/technologies/nfv

[16] H. L. Mai, N. T. Nguyen, G. Doyen, R. Cogranne, M. Wissam, E. Montes de Oca, and O. Festor, "Towards a security monitoring plane for named data networking: Application to content poisoning attack," in *Network Operations and Management Symposium (NOMS), 2018 IEEE*. IEEE, 2018.

[17] D. Kondo, T. Silverston, H. Tode, T. Asami, and O. Perrin, "Name anomaly detection for icn," in *Local and Metropolitan Area Networks (LANMAN), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–6.

[18] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[19] "Implementation of content poisoning attack detection and reaction in virtualized NDN networks," in *21st Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2018, Paris, France, February 19-22, 2018*, 2018, pp. 1–3. [Online]. Available: https://doi.org/10.1109/ICIN.2018.8401591