Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 3, 7956 – 7963

Application Aware Self Adaptive Decentralized Management In Software Defined Networking

[1]boggula Rojaramani, [2]bathula Sowjanya, [3]sirisha Kamsali, [4]dr K Sreenivasulu

[1,2,3]Assistant Professor, Department of CSE, G.Pullaiah college of Engineering and Technology (Autonomous), Kurnool, A.P, India
[4]Professor, Department of CSE, G.Pullaiah college of Engineering and Technology (Autonomous), Kurnool, A.P, India

Abstract

The Software Defined Networking (SDN) will enable the network programmability realization and application-aware networking in network devices based on the separation of data and control planes. While developing the management architectures based on SDN, this has significant importance for designing a system which will offer consistent updates in real time for the management applications of heterogeneous systems. The management system should scale as the network size grows, minimize the overhead of available resources, provide an accurate network view, and support vacuum applications with low latency requirements. As a result, self-adaptive distributed management will be introduced for applications in software-defined networks. Management functions will be implemented on a case-by-case basis. This concept has the potential to introduce a new node, namely the SDN manager, into the SDN architecture. This presented approach is evaluated based on realistic use case conditions, and a test bed will be set up to compare performance between scenarios with and without mapping, which can show that mapping between network resources and applications results in a significant improvement in performance. This demonstrated management approach will have strong impact on SDN architecture evolution since presented distributed execution filed will be a generic one, so this will be utilized for management as well as application or control functions distribution.

KEYWORDS: network management, Software-Defined Networks, Self-adaptation, application-aware networking.

I. INTRODUCTION

Traditional network management methods have been challenged by advanced SDNs in recent years [1]. SDN technologies are emerging as one of the most promising solutions for simplifying and improving operator tasks, as they will enable the development of applications that will automatically reconfigure the network [2]. The SDN is aiming for providing network

programmability through a unified control plane of network. Controllers in current SDN solutions may be able to expose open APIs (Application Programming Interface) via a service abstraction. For example, an application may be able to use a service plug-in that is distinct from a single controller to call a connection service that spans multiple domains [3].

VTN (Virtual Tenant Network) Coordinator applications can build virtual networks on top of the underlying physical network [4]. You can maintain an underlying connection or call network resources by adding a set of flow table entries to your physical network via the controller. This is in sharp contrast to the traditional network management concept, in which network provisioning is done manually or by OSS (Operations Support Systems) and is invisible to the application. Mean while in current SDN implementations still the resource management is hugely developed for the purpose of conventional network management and most part of the controller is not known that how applications use the resources of network. One of the primary requirements for any kind of network management system is efficient resource monitoring [5].

Generally the role of management is to handle the failures, provide devices configuration, offer billing, network performance optimization, network handling and security of devices (namely functionality of FCAPS (Fault, Configuration, Accounting, Performance And Security)). In previous networks the management can be centralized. The management solution complexity is most popular and the role of human is vital. The operations are divided between control planes and management based on the type of operation and the frequency with which events occur. When the event appears often then it can be handled through control plane. When its occurrence will be rare (basically in reconfiguration and setup of network) such kind of mechanism or event is normally belongs to management plane. Though, strictly defined border line is not there between these planes. In addition in certain concepts of cross layer the control and management would cooperate for solving the issues such as faults, etc. The control and management planes will be utilized different protocols.

The new SDN-based network control and management architecture that can support the applications of dynamic resources management in fixed back-bone architectures [6]. In fact, multiple applications can use the network, each with a Different Service Level Agreement (SLA) such as availability and packet loss rate. It is important to understand how each application uses the same network. As an instance, the network failure disrupts the number of applications that are passing through the failure point. For fast realization and recovering of differentiated failure, one should aware precisely about the correspondence between affected applications and resources of network [7]. The awareness about applications in SDN results more feasibility for future networks. The SDN is extended to be aware of application has gained more attention from industries and academia. From the observations one thing is ignored in SDN based application awareness that the usage applications network resources by the usage must be recorded in controller resource database management.

The set of LMs (Local Managers) can distribute on the network, various host MAs (Management Applications) which will implement the desired logic for deciding over reconfigurations of network. The MAs will instantiate on LMs as modules embedded data models and running on the basic execution platform provided by LMs. Every MA is executed in all LMs or in the subset of LMs (one is operating at network nodes edge). The LMs make configuration decisions, which are translated as a set of commands and transmitted to the hardware via a southbound interface (e.g., OpenFlow) that will indicate the actions sequence to be enforced in order to update network parameters.

The paper structure is following: Section II relates the literature survey, Section III introduces the described methodology, the results of the experiment setup and evaluation are shown in Section IV, and the paper is concluded in Section V.

II. LITERATURE SURVEY

B. Chatras et. al. [8] has introduced the NFV (Network Functions Virtualization) enabled network slicing for 5G. The selected function can route the telecommunications towards a core network slice which is suitable to provide a specific service in which the criteria of definition to access core network and slice includes the desires for satisfying the requirements of communication as well as application. With this approach, each core network slice can be configured with at least one NF (Network Feature), and a particular network feature within the slice is used by multiple slices, eliminating the need for repeated builds. Moreover other features of network can be utilized for particular slices.

Liao et al. [9] presented SDN-based energy-efficient data collection strategy for WSN (Wireless Sensor Networks). In this approach, sensor collects the information and it is given as a data attributes like temperature, etc. then the sensor may compare either the processing technique records the temperature data in the flow table or not. When data is recorded then the data can be processing as per the rule. If no data is recorded, the sensor will request that the flow table be updated by the controller. A dynamic flow table variation with area is used to avoid additional congestion and broadcasting in the SDN server. This approach is evaluated by the computation of metric like consumption of energy to store and retrieve the data changing sensing events.

A. Blenk et. al. [10] has presented the core cloud is comprised of virtualized servers, as well as virtual routers (virtual machine monitors) and virtual switch functions on the servers. In this case, the SDN controller's primary responsibility is to establish an SDN channel between each virtual core. Over an MPLS-L3 VPN (Multiprotocol Label Switching- L3 Virtual Private Networks) router, a cloud server and a cloud Data Centre Gateway (DCG / W) are connected. The SDN controller is then activated to perform the necessary concatenation of the SDN channel and the MPLSL3VPN. Edge clouds are linked to the core cloud via an IP/MPLS (IP Multi-Protocol Label Switching) backbone network and can be divided into network slices for specific applications. Luo et al. [11] presented an efficient energy approach for improving Industrial WSNs (IWSNs) applications using NFV and SDN. The new framework is utilized where a controller is introduced, workflow and algorithm is elaborated for various segments. The controller can include various parts like selection of nodes, pre-computing, selection of cluster heads, selection of relay nodes and formation of cluster, distribution of VNF (Virtual Network Functions) instances and flow tables to the nodes of sensor. This approach is validated with the simulations while comparing the two most popular classical protocols.

Tang et al. [12] presented the mobility management model based on control and data plane separation. The resource handler is designed for maintaining the information of location, rendezvous point is utilized for forwarding the data packets and Interest. This approach does not avoid the single point of failure issue because all the data, control packets are transmitted via rendezvous point and resource handler.

Kuklinski and Chemouil et. al. [13] has described the SDN management challenges while comparing it with the classical networks. Started with FCAP (Fault, Configuration, Accounting, Performance And Security) approach, the authors exhibited OpenFlow various functionalities and limitations majorly its inabilities for performing the programmability of controller, therefore a novel management function will be required in SDN. S. Sasidharan et. al. [14] noticed that SDN presents the opportunities for simplifying the management by offering the central control/data acquisition entity and it can also create the extra requirements corresponds to the programmability of network, network bootstrapping and dynamism in policy creation.

Bari et al. [15] presented a Policy Cop which is an autonomic enforcement framework of QoS (Quality of Service) policy for SDN. This framework can allow for the implementation and enforcement of QoS SLAs (Service Level Agreements) in an OpenFlow-based network. Similar to OpenSec, PolicyCop employs a step-by-step approach to converting policies into flow rules. However, rather than QoS violations, the primary focus is on responding to network security alerts.

III. APPLICATION AWARE SELF ADAPTIVE DECENTRALIZED MANAGEMENT IN SDN

Architecture of Application Aware Self Adaptive Decentralized Management in Software Defined Networking is represented in below Fig. 1. For the management of decentralized SDN, the presented model employs a separate node, namely the SDN manager. This concept's applicability in SDN networks is one of its key benefits. Despite the presence of the SDN manager in this variant, certain management functions, particularly those that can be combined with controller operations, are implemented in the controller. The SDN Manager communicates with the operator console, acts as a repository for controller software modules, and is linked to the OSS/BSS (Operations Support System and Business Support System) system.

The SDN manager is more important in this model than in the previous model. It will be in charge of all network node software maintenance, including controller software modules and switch management software. This distributed management model key feature is SDN manager abilities to transmit and execute a software module in each node of network. One of the interesting features of this variant is the ability to launch protocols and nodes used to manage or control SDN networks.

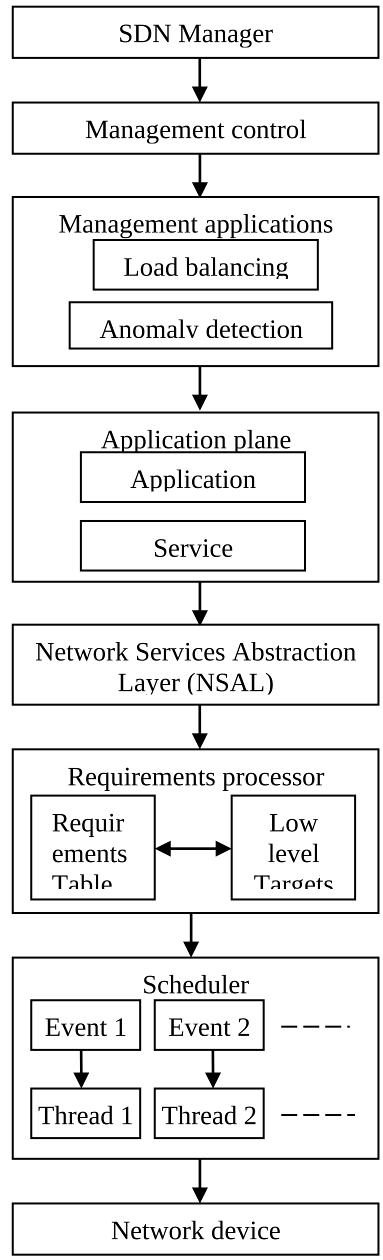

Fig. 1: ARCHITECTURE METHODOLOGY

An HL (High Level) target is a list of monitoring task targets (high-level identifiers in an abstract view your application's network), such as: A list of appropriate paths for path usage requirements. Mon can be a single parameter, such as an explicit sequence of polling intervals or sampling intervals. For example, based on the analysis of the measured parameters, the regulator can modify the monitoring requirements. SAM is a Self-Adaptive voting Mechanism that adjusts polling rates in real time based on network traffic behavior. When this mechanism is in place, the MA individual must adhere to the measurement rate limit. Under the M_{on} Times attribute, specify the acceptable monitoring frequency interval [fmin, fmax].

Through the Network Service Abstraction Layer (NASL), controllers can provide open APIs For example; an application can use a single controller with multiple service plug-INS to invoke connectivity services across multiple domains.

The component's next step is a conversion routine based on task specifications that maps each new table entry to one or more low-level targets. Each Low-Level target (LL target) is capable of identifying a specific physical resource. It is possible to map a switch port or a set of flow rules. A subset of the switch flow table, to be precise. These entities are saved in the following format in the low level target table: h LL target, Op type, [Requirement id], Sched state I The Op type specifies the type of measurement operation to be performed, such as gathering the average traffic rate of a particular switch interface. [Req id] is a list of application requirement references that can be reverse-translated. The Sched state flag indicates whether the low-level target refers to a new monitoring requirement (i.e., measurement operations must be scheduled from the ground up) or to a previously completed task that must be modified (i.e., operations have to be re-scheduled).

The Scheduler component is responsible for creating and managing individual measurement procedures that are executed as threads (e.g. those that require a single message exchange with a switch). Each insert into the low-level target table, as well as any modification of existing tuples containing the measured time, triggers this function. Randomly scheduling new metrics may result in none of them receiving the appropriate conversion resources. Therefore, when the scheduler is called, it runs an admission control procedure to determine whether new low-level target measurement procedures can be performed based on the current measuring load.

The Scheduler component creates and manages individual measurement procedures (for example, those requiring the exchange of unique messages with a switch) that are executed as threads. This function is called for each insert into the table of low-level targets, as well as for any modification of existing tuples containing bar times. When new metrics are randomly scheduled, none of them can get enough conversion resources. Therefore, when the scheduler is called, it runs an admission control process to determine if measurement activities for the new low-level target can be completed with the current measurement load.

IV. RESULT ANALYSIS

Open Daylight is used as the controller software, and as a plug-in for Open Daylight and created a mapping module. The controller host is an HP ProLiant ML350 Gen9 server that is linked to an HP (Hewlett- Packard) Inspiron 560 computer that runs Mininet and is used to generate virtual switches for the test topology.

The test network is made up of 500 network components. A route consists of ten network elements, each of which contains 100 redundant flow table entries. The number of distinct apps on the network has been increased from 100 to 600. Because the controller forwards flow table entries to network devices for application connectivity, searching for the desired flow table entries in networks with many applications takes a long time. However, as the number of applications increases, the fixed depth of Application Aware Self-Adapting Decentralized Management (AASADM), also known as the mapping module, will maintain the time-cost curve time.

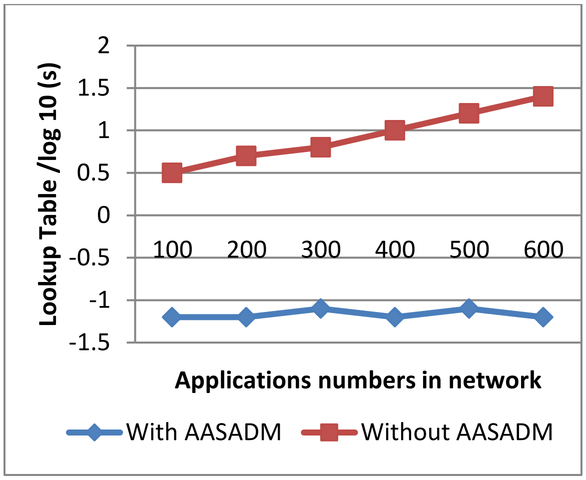

Fig. 2: TEST RESULTS

Hence, consider the distributed management system, distributed program instances can make decisions based on data collected from remote locations. This subsection describes the impact of monitoring information distribution among LMs (Local Managers) on the performance of LB (Load Balancer) applications and the impact on the provision of cloud gaming services and fully distributed deployment test. Congestion episodes occur on a particular connection 1 under the control of a specific Load Balancer (LB) instance, but the decision to offload is made by another application instance outside of the local partition set (LB remote).

Despite being primarily focused on SDN management, the presented notion appears to be a key step toward more advanced SDN. It can be utilized to address some of SDN's core issues, such as there are circumstances where data can be successfully processed at the network edge despite scalability and other constraints associated with centralized only network control. The presented concept adds a new component to the SDN architecture: the SDN manager. The presence of such a node in the SDN architecture is justified for a variety of reasons.

The experiments were performed on two network topologies. Topo1 has an average link latency of 5ms, with end-to-end (round-trip) latency ranging from [25ms to 70ms]. Link latency on Topo2 is artificially calibrated to allow us to experience increasing delays from start to finish, with latency for the whole trip between 100 and 150 milliseconds. Clients are distributed across five user locations in both topologies. The LM positions are given as input, the number of associated hops and the delay between pairs of LMs are computed.

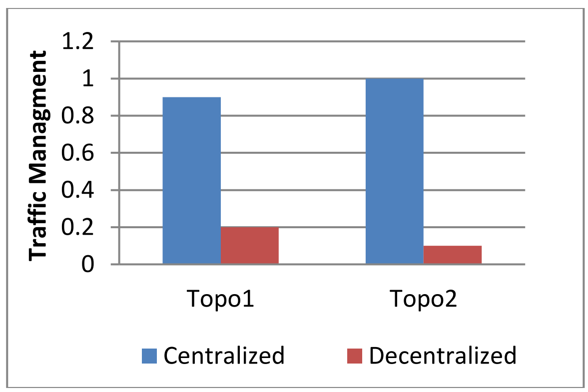

Fig. 3: PERFORMANCE OF THE DECENTRALIZED MANAGEMENT APPROACH

compared to the centralized situation, a considerable delay decrease for minimal distance in the decentralized configurations has seen. Topo1 has a discount of up to 57 percent, and Topo2 has a discount of even more (61 percent). Topo2 paths typically have a higher hop count and a higher latency. The maximum range delay reduction is small, up to 17 percent for Topo1 and up to 40 percent for Topo2. As expected, the higher the percentage of reconstruction calculated near where important insights are gained, the lower the loop delay achieved by the distributed approach.

V. CONCLUSION

In this paper describes application-aware self-adaptive distributed management in software-defined networking design. This approach is based on a distributed architecture that meets the needs of large networks with geographically dispersed devices. Based on realistic topologies and advanced use case services, the distributed monitoring framework can significantly reduce control loop delays, especially if most of the reconfiguration decisions are made near the site. It shows that the responsiveness of the reconstruction can be improved. It will demonstrate the performance of common application-aware network operations with and without such modules using Open Daylight as an example. The fixed depth of Application-Aware Self-Adaptive Distributed Management (AASADM), also known as the mapping engine, stabilises the time and cost curves as the number of applications grows.

VI. REFERNCES

[1] B Harshitha, K.N Madhusudhan, S Lalitha, “Global Connectivity in Software Defined Network”, 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Volume: 1, Year: 2021

[2] S.K. Nivetha, N. Senthilkumaran, P. Suguna, T. Vanaja, “Analysing the Performance of Software Defined Network Controllers with Various Topologies”, 2021 International Conference on Computer Communication and Informatics (ICCCI), Year: 2021

[3] Dorabella Santos, João P. Vidal, Teresa Gomes, Lúcia Martins, “A Heuristic Method for Controller Placement and Enhanced Availability between SDN Controllers”,2020 11th International Conference on Network of the Future (NoF), Year: 2020

[4] Reinhard Frank, Matthias Scheffel, Florian Zeiger, Hans-Peter Huth, “Enabling Multi-Tenant Networks for the Automation Industry”,2019 International Conference on Networked Systems (NetSys), Year: 2019

[5] Mohamed Abderrahim, Meryem Ouzzif, Karine Guillouard, Jérôme François, Adrien Lebre, Charles Prud'homme, Xavier Lorca, “Efficient Resource Allocation for Multi-Tenant Monitoring of Edge Infrastructures”, 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Year: 2019

[6] Beny Nugraha, Mehrdad Hajizadeh, Trung V. Phan, Thomas Bauschert, “A Novel Impact Analysis Approach for SDN-based Networks”, 2019 IEEE Conference on Network Softwarization (NetSoft), Year: 2019

[7] Teddy Edu-yaw, Eric Kuada, “Service Level Agreement Negotiation and Monitoring System in Cloud Computing”, 2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST), Year: 2018

[8] B. Chatras, U. S. T. Kwong, N. Bihannic, “NFV enabling network slicing for 5G,” IEEE Conference Publication from Innovations in Clouds, Internet and Network(ICIN), 2017 20th Conference on, March, 2017.

[9] W. Liao and S. Kuai, "An Energy-Efficient SDN-Based Data Collection Strategy for Wireless Sensor Networks," 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), Kanazawa, 2017, pp. 91-97

[10] A. Blenk, A. Basta, M. Reisslein and W. Kellerer, “Survey on Network Virtualization Hypervisors for Software Defined Networking,” in IEEE Communications Surveys & Tutorials,vol. 18, no. 1, pp. 655-685, Firstquarter 2016.

[11] S. Luo, H. Wang, J. Wu, J. Li, L. Guo and B. Pei, "Improving Energy Efficiency in Industrial Wireless Sensor Networks Using SDN and NFV," 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, 2016, pp. 1-5.
[12] J. Tang, H. Zhou, Y. Liu et al., “A Source Mobility Management Scheme in Content-Centric Networking,” in Proc. 11th IEEE Consumer Communications and Networking Conference (CCNC), pp.176-181, Jan 2014
[13] S. Kuklinski and P. Chemouil, "Network management challenges in software-defined networks," IEICE Trans. Commun., vol. 97, pp. 2-9, 2014
[14] S. Sasidharan and S. K. Chandra, "Defining future SDN based network management systems characterization and approach," in International Conference On Computing, Communication and Networking Technologies (ICCCNT), 2014, pp. 1-5.
[15] M. Bari, S. Chowdhury, R. Ahmed, and R. Boutaba, “PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks,” in IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, November 2013.