

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: December 23, 2017

K. Wehmuth  
A. Ziviani  
LNCC  
June 21, 2017

A Reference Model for Representing SDN Environments  
draft-wehmuth-nmrg-sdn-model-00

Abstract

Software-Defined Networks (SDNs) are multilayer systems. In this context, this draft defines a graph-based reference model capable of properly representing such complex multilayer networks. The defined reference model thus eases the management and planning of SDN environments.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 23, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Why modeling SDNs as multilayer networks . . . . .	3
3. How to model a SDN as a multilayer network . . . . .	4
3.1. Introduction to MultiAspect Graphs . . . . .	4
3.2. Multilayer graph (MLG) definition . . . . .	4
3.3. Algebraic representations and structures . . . . .	5
3.4. MLG adjacency matrix . . . . .	5
3.5. SDN reference model . . . . .	8
4. Conclusion . . . . .	9
5. IANA Considerations . . . . .	9
6. Security Considerations . . . . .	9
7. Informative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Introduction

Software-Defined Networks (SDNs) are inherently multilayer systems. In addition to the traditional layers associated with the separated data and control planes, other layers can be considered to support structures, such as hierarchical controllers, structured interaction between applications, use of Network Functions Virtualization (NFV) on SDN environments, among others. It is important to properly represent such a complex structure in a convenient way that allows modeling and analysis of a SDN environment with a single object.

In this context, we propose the use of a theoretical graph framework [Wehmuth2016], capable of modeling multilayer complex networks, for representing SDN environments. This framework is capable of representing complex networks containing an arbitrary (finite) number of layers, thus allowing the representation of SDN systems with any number of associated layers. In this framework, if desired, the usual SDN layers can be divided into sub-layers allowing the creation of more detailed and structurally rich SDN reference models. Therefore, this framework is capable of modeling various distinct SDN architectures, such as ForCES [RFC3746], SDN systems adherent to [RFC7426], [draft-irtf-sdnrg-pop-00], or any other layered networking architecture. Further, the considered framework has the property of guaranteeing that any model created in it is necessarily equivalent (isomorphic) to a directed graph. Therefore, all knowledge available for directed graph analysis can be directly applied to representation based on this framework. Additionally, the graph theoretical knowledge can be extended in the framework in order to allow for results based on advanced aggregation of layers that are present on the represented models.

Since SDN reference models created using the proposed framework are guaranteed to be equivalent to directed graphs, they can be represented in their canonical compact form, or by means of matrices usually employed for graph representation (e.g., adjacency matrices). Further, well-known graph algorithms can be applied directly to the representation based on the considered framework, allowing for the straightforward computing of distances among objects on a SDN system, the evaluation of the flow capacity of any given path on the system, the finding of structurally relevant objects or edges in the system (i.e. centrality evaluation), the construction of flow matrices, or any other operation possible for directed graphs.

The proposed SDN reference model fully reflects the complexity of SDN systems, while also allowing the straightforward usage of the model as a directed graph. Moreover, the fact that the whole network structure can be represented by a single mathematical object greatly contributes to the consistency of the obtained results. Therefore, the proposed framework can be useful either in an offline environment, where it can be used for system design and simulation of what-if scenarios, or as an online environment deployed, for instance, in the SDN controller(s), allowing for real-time evaluation of the structural properties of the whole system network. The proposed reference model for representing SDN environments thus contributes to the management and planning of these environments.

## 2. Why modeling SDNs as multilayer networks

Since SDNs are intrinsically layered systems, it is natural to model it as a multilayer network. Moreover, the usage of such a model has the advantage of clearly exposing the SDN layered structure. In a multilayer model, not only the natural layers visible on a SDN are clearly represented, but also, if desired, it is possible to divide each SDN layer into a set of sub-layers. In this way, structures such as hierarchical distributed control architectures, where multiple controllers with distinct hierarchy can be allocated to distinct control sub-layers. In this manner, not only the topological structure of the controllers is clearly modeled, but also, their hierarchical structure. Further, structures that may sometimes be attached to a SDN system, such as NFVs, can be modeled in layers specifically reserved for them, making the whole structure clear.

Moreover, by modeling a SDN as a multilayer network, it becomes possible to take advantage from the body of knowledge already established in graph theory for analyzing the SDN structure.

### 3. How to model a SDN as a multilayer network

#### 3.1. Introduction to MultiAspect Graphs

A MultiAspect Graph (MAG) is a graph generalization introduced in [Wehmuth2016] that is shown to be equivalent to a directed graph. In this generalization, the set of vertices, layers, time instants, or any other independent features are considered as an aspect of the MAG. For instance, a MAG is able to represent multilayer or time-varying networks, while both concepts can also be combined to represent a multilayer time-varying network and even other higher-order networks. Since the MAG structure admits an arbitrary (finite) number of aspects, it hence introduces a powerful modeling abstraction for networked complex systems.

#### 3.2. Multilayer graph (MLG) definition

We propose to model SDN systems by using a Multilayer Graph (MLG) model, that is a particular case of a MultiAspect Graph (MAG) [Wehmuth2016], in which the vertices and layers are the key features (i.e., aspects) to be represented by the model. Formally, a MAG can be defined as an object  $H=(A,E)$ , where  $E$  is a set of edges and  $A$  is a finite list of sets, each of which is called an aspect. In our case, for modeling a MLG, we have two aspects, namely vertices and layers, i.e.  $|A|=2$ . For the sake of simplicity, this 2-aspect MAG can be regarded as representing a MLG with an object  $H = (V, E, L)$ , where  $V$  is the set of vertices,  $L$  is the set of layers, and  $E$  is a subset of  $(V \times L \times V \times L)$ , that is the set of edges. As a matter of notation, we denote  $V(H)$  as the set of all vertices in  $H$ ,  $E(H)$  the set of all edges in  $H$ , and  $L(H)$  the set of all layers in  $H$ .

An edge  $e$  in  $E(H)$  is defined as an ordered quadruple  $e = (u, l_a, v, l_b)$ , where  $u, v$  in  $V(H)$  are the origin and destination vertices, while  $l_a, l_b$  in  $L(H)$  are the origin and destination layers, respectively. Therefore,  $e = (u, l_a, v, l_b)$  should be understood as a directed edge from vertex  $u$  at layer  $l_a$  to vertex  $v$  at layer  $l_b$ . If one needs to represent an undirected edge in the MLG, both  $(u, l_a, v, l_b)$  and  $(v, l_b, u, l_a)$  should be in  $E(H)$ .

An edge  $e = (u, l_a, v, l_b)$  in our model may be classified into four classes depending on its characteristic:

- o Intralayer edges connect two vertices in a same layer,  $e$  is in the form of  $e = (u, l_a, v, l_a)$ , where  $u$  and  $v$  are distinct;
- o Interlayer edges connect the same vertex in two distinct layers,  $e$  is in the form of  $e = (u, l_a, u, l_b)$ , where  $l_a$  and  $l_b$  are distinct;

- o Mixed edges connect distinct vertices in distinct layers,  $e$  is in the form of  $e=(u, l_a, v, l_b)$ , where  $u$  and  $v$  are distinct and  $l_a$  and  $l_b$  are distinct;
- o Intralayer self-loop edges connect the same vertex in the same layer,  $e$  is in the form of  $e=(u, l_a, u, l_a)$ .

Further, we define a composite vertex as an ordered pair  $(u, l_a)$ , where  $u \in V(H)$  and  $l_a \in L(H)$ . The set  $VL(H)$  of all composite vertices in a MLG  $H$  is given by the Cartesian product of the set of vertices and the set of layers, i.e.  $VL(H) = V(H) \times L(H)$ . As a notation note, a composite vertex is represented by the ordered pair that defines it, e.g.  $(u, l_a)$ , where  $u \in V(H)$  and  $l_a \in L(H)$ .

### 3.3. Algebraic representations and structures

In this section, we discuss ways to properly represent a MLG using our proposed model. Similarly to static graphs, a MLG can be fully represented by an algebraic structure, like the MAG structure from which our MLG model is derived. In this work, we adopt matrix-based representations, in particular the adjacency matrix.

In order to illustrate such representations, we use the MLG  $W$  presented in Figure 1.

### 3.4. MLG adjacency matrix

Since every MAG has a directed graph that is equivalent to it, the same holds for our MLG model, since it is a particular specialized case of a MAG. Consequently, it follows that the MLG can be represented by an adjacency matrix. For the sake of standardization and without loss of generality, we define that in a MLG the first aspect represents the vertices (i.e. the objects that compose the SDN system) and the second aspect represents the layers of the represented system.

In the more general environment represented by a MAG, a companion tuple is used in order to properly identify and position each composite vertex of the equivalent graph in the adjacency matrix. Since the case we present in this work is restricted to MAGs with 2 aspects, it follows that the companion tuple is reduced to a pair, which in the first entry has the number of vertices and the second entry has the number of layers. For instance, considering the MLG example of Figure 1, the companion tuple associated with its adjacency matrix is  $(10,3)$ , since there are 10 vertices and 3 layers. The function of the companion tuple is only to ensure that the order by which the composite vertices are placed in the adjacency matrix is the one shown in Figure 2. Since in the case where the number of

aspects is restricted to 2 this placement can be easily achieved, in this work we do not further mention the companion tuple.

To get the MLG adjacency matrix, we only need to consider that each composite vertex  $(u, la)$  can be thought of as a vertex in a directed graph. This directed graph has  $|V| * |L|$  vertices and, as a consequence, its adjacency matrix has  $|V| * |L| * |V| * |L| = |V|^2 * |L|^2$  entries. Since the non-zero entries of this matrix correspond to the edges of the MLG, further analysis show that this matrix is usually sparse and can therefore be stored in an efficient way.

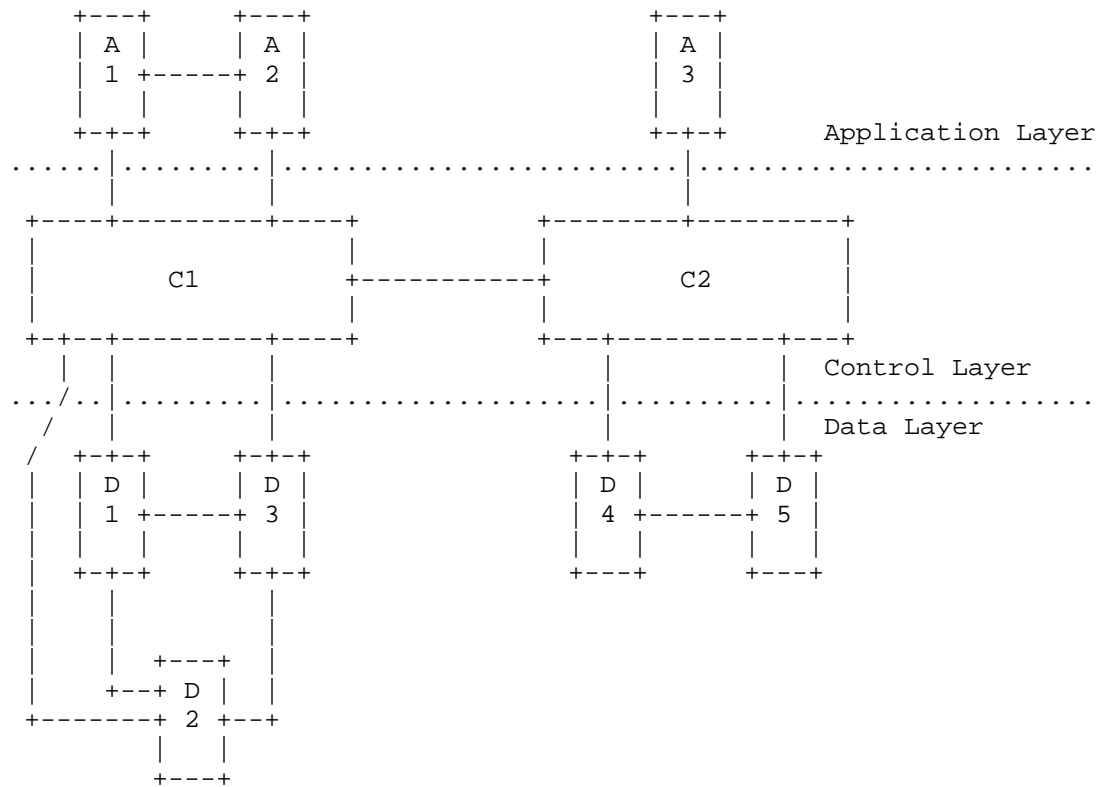


Figure 1: SDN Example

Figure 2 shows the adjacency matrix obtained for the illustrative MLG W shown in Figure 1. From Figure 2, we highlight that the adjacency matrix form of the MLG has interesting structural properties.

+-		+-	
0 1 1 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D1
1 0 1 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D2
1 1 0 1 0 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D3
0 0 1 0 1 0 0 0 0 0	0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0 0	D4
0 0 0 1 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0 0	D5 Data
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	C1 Layer
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	C2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A3
.....			
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D3
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D4
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D5 Ctrl
1 1 1 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 1 1 0	C1 Layer
0 0 0 1 1 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 1	C2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	A3
.....			
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D3
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D4
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	D5 Apps
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	C1 Layer
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	C2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0	A1
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 0	A2
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	A3
+-		+-	
D D D D D C C A A A	D D D D D C C A A A	D D D D D C C A A A	
1 2 3 4 5 1 2 1 2 3	1 2 3 4 5 1 2 1 2 3	1 2 3 4 5 1 2 1 2 3	
Data	Control	Apps	
Layer	Layer	Layer	

Figure 2: SDN Matrix Example

First, each one of the ten vertices (identified as D1, D2, D3, D4, D5, C1, C2, A1, A2 and A3) of the MLG W clearly appears as a separate entity in each of the three layers (l0 - Data, l1 - Control, and l2 - Applications) that compose the MLG W. Second, the main block diagonal contains the entries corresponding to the intralayer edges of each layer. In these blocks, the entries corresponding to the

intralayer edges of the MLG carry value 1. Finally, the entries at the off-diagonal blocks correspond to the interlayer edges. The eight interlayer edges present at the MLG  $W$  are indicated by the value 1 on the off-diagonal blocks. Further, we remark that all these structural properties derive from the order adopted for representing the vertices and layers present in the MLG and can be readily verified in the matrix form in a quite convenient way.

### 3.5. SDN reference model

From the MLG definition, it follows that a MLG can represent multilayer networks with an arbitrary (finite) number of layers. At a first glance, this would be enough to represent a multilayer system, such as a SDN. However, additional definitions can be made in order to provide a clear description of a SDN. For instance, a SDN reference model could benefit from an adequate name structure for its layers.

We start by naming the four basic layers considered in this work as  $L_d$  for the data layer,  $L_c$  for the control layer,  $L_a$  for the application layer, and  $L_n$  for the NFV layer. Further, each basic layer can be defined in a number of sub-layers, yielding  $L_{d1}$  to  $L_{dj}$  for data plan layers,  $L_{c1}$  to  $L_{ck}$  for control plan layers,  $L_{a1}$  to  $L_{am}$  for application layers and  $L_{n1}$  to  $L_{ni}$  for NFV layers. In this way, the total number of layers in the SDN model is given by  $|L| = j + k + m + i$ . Note that not all layers need to be necessarily represented. For instance, a simple SDN with 1 data plan layer, 1 control plan layer, 1 application layer, and no NFV layer, can be modeled by a 3 layer MLG, where  $j = k = m = 1$  and  $i = 0$ .

We remark that since a MLG is equivalent to a directed graph, all extensions usually applied to graphs, such as edge weights and vertices weights can be directly applied to MLGs, and also, all algorithms known for directed graphs can be directly applied to MLG.

In addition to the traditional directed graph algorithms, it is possible to construct algorithms that use the full information present on the MLG and deliver aggregated results (e.g. results for vertices; disregarding layers). By using these algorithms, the results do not consider the artifacts generated by the traditional aggregation operation. This means, for instance, that aggregated paths are calculated using only paths that are actually present on the MLG.



#### 4. Conclusion

In this work, we presented a SDN reference model based on MLGs, which are a special case of a MultiAspect Graph (MAG). In particular, a MLG is a MAG with exactly 2 aspects, named vertices and layers. Since the MLG has a fix number of aspects, it can be constructed with a simpler structure than a MAG.

We show that a MLG can properly represent a SDN system and that since the MLG inherits the basic properties of a MAG, in particular, the equivalence (isomorphism) to directed graphs, the knowledge present in the theory of directed graphs can be applied to our proposed reference model for representing SDN environments. This makes our model a convenient way of representing a SDN, by both expressing it as a multilayer system, while also providing a well established theoretical ground and available algorithms to build analytics.

#### 5. IANA Considerations

This memo includes no request to IANA.

#### 6. Security Considerations

Similarly to [RFC7426], this document does not propose a new network architecture or protocol and therefore does not have any impact on the security of the Internet. However, security in SDN environments is discussed in the literature, e.g. in [SDNSec], [SDNSecSrv], and [SDNSecOF].

#### 7. Informative References

[Wehmuth2016]

Wehmuth, K., Fleury, E., and A. Ziviani, "On MultiAspect graphs", Theoretical Computer Science Vol. 651, pp. 50-61, DOI 10.1016/j.tcs.2016.08.017, October 2016.

[SDNSecOF]

Kloti, R., Kotronis, V., and P. Smith, "OpenFlow: A Security Analysis", 21st IEEE International Conference on Network Protocols (ICNP) pp. 1-6, October 2013.

[SDNSecSrv]

Scott-Hayward, S., O'Callaghan, G., and S. Sezer, "SDN Security: A Survey", In IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1-7, 2013.

## [SDNSec]

Kreutz, D., Ramos, F., and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks", In Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking, pp. 55-60, 2013.

## [I-D.irtf-sdnrg-pop]

Tian, Y., "Programming Model for Protocol Oblivious Forwarding SDN Networks", draft-irtf-sdnrg-pop-00 (work in progress), January 2017.

## [RFC3746]

Yang, L., Dantu, R., Anderson, T., and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, DOI 10.17487/RFC3746, April 2004, <<http://www.rfc-editor.org/info/rfc3746>>.

## [RFC7426]

Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<http://www.rfc-editor.org/info/rfc7426>>.

## Authors' Addresses

Klaus Wehmuth  
LNCC  
Avenida Getulio Vargas, 333  
Petropolis, RJ 25651-075  
Brazil

Phone: +55 24 2233-6000  
Email: klaus@lncc.br

Artur Ziviani  
LNCC  
Avenida Getulio Vargas, 333  
Petropolis, RJ 25651-075  
Brazil

Phone: +55 24 2233-6199  
Email: ziviani@lncc.br