



HAL
open science

Metarouting with Automatic Tunneling in Multilayer Networks

Noureddine Mouhoub, Maria Moloney, Damien Magoni

► **To cite this version:**

Noureddine Mouhoub, Maria Moloney, Damien Magoni. Metarouting with Automatic Tunneling in Multilayer Networks. *Journal of Network and Computer Applications (JNCA)*, 2024, 230, pp.103929. 10.1016/j.jnca.2024.103929 . hal-04621119v2

HAL Id: hal-04621119

<https://hal.science/hal-04621119v2>

Submitted on 2 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Research paper

Metarouting with automatic tunneling in multilayer networks[☆]Noureddine Mouhoub^a, Maria Moloney^b, Damien Magoni^{a,*}^a LaBRI-CNRS, University of Bordeaux, Talence, 33405, France^b PEL, University College Dublin, Belfield, Dublin 4, Ireland

ARTICLE INFO

Keywords:

Metarouting
 Routing algebra
 Partial order
 Multilayer network
 Automatic tunneling
 Path computation

ABSTRACT

Metarouting allows for the modeling of routing protocols using an algebraic structure called routing algebra. Routing protocols requiring design or validation can easily be modeled using this approach. To date, however, existing research on routing algebras has mainly focused on applying this approach to routing protocols that are generally used in networks which have a single addressing and forwarding protocol. The basic algebraic structures used in such contexts are semirings, Sobrinho's algebras and algebras of endomorphisms. In this paper, we propose the modification of these existing routing algebras to deal with networks that contain multiple forwarding protocols where tunnels are omnipresent. To achieve this, we define new algebraic structures derived from the three aforementioned ones, in order to model the generalized routing problem with automatic tunneling entitled *valid paths* algebra. All of our routing algebras are defined as semi-direct products of two structures, the well-known *shortest paths* algebra and the proposed *valid paths* algebra. These new algebras are isotonic and non-monotonic with a partial order. We propose a fixed point for those new algebras and we prove the iterative convergence to the optimal solution of the *valid shortest paths* problem.

1. Introduction

Routing algebras for classical routing protocols can be dated as far back as 1971. It was not until 2003, however, that they were studied in the context of Internet routing protocols such as BGP. In fact, the problem of automatic tunneling management for multilayer network routing has itself only been tackled since 2009 and has since attracted increasing attention due to the advent of IPv6 (Raste and Kulkarni, 2008) and the deployment of VPNs. From this perspective, leveraging routing algebras to derive properties on multilayer networks becomes compelling. Our aim in this paper is to define such algebras and prove some properties related to network routing with automatic tunneling. Our contribution can be summarized as follow:

- We define three routing algebras for performing path computation with tunnels: one as a semiring, one as an algebra of functions, and one as a Sobrinho's algebra.
- We show that these routing algebras are isotonic and non-monotonic with a partial order.
- We propose a fixed point for these new algebras.

- Finally, we prove the iterative convergence to the optimal solution of the *valid shortest paths* problem called *vsp*.

This paper is an extended version of a previous short paper by Mouhoub et al. (2022c) presented at the Workshop on New IP and Beyond held in conjunction with the International Conference on Network Protocols. The short paper only defined the semiring algebra and did not include any proofs. In this extended version, we propose and define two more routing algebras, as well as proving that all these algebras are isotonic and not monotonic. Finally, we prove the convergence to an optimal solution of the length of elementary multilayer paths in any multilayer network.

This paper is organized as follows: Section 2 reviews research on routing algebras and multilayer routing. Section 3 presents the algebraic foundations used throughout the paper, while Section 4 details the multilayer network model on which we extend our routing algebras. This is followed in Section 5 by three definitions for algebraic structures (semiring, algebra of functions, and algebra of Sobrinho) needed for the valid path problem. Finally, we prove a theorem in Section 6 which

[☆] This work was funded by The French National Research Agency - HÉRA project. Grant no.: ANR-18-CE25-0002.

* Corresponding author.

E-mail addresses: noureddine.mouhoub@u-bordeaux.fr (N. Mouhoub), maria.moloney@ucd.ie (M. Moloney), damien.magoni@u-bordeaux.fr (D. Magoni).

URLs: <https://www.labri.fr/perso/nmouhoub/> (N. Mouhoub), <https://www.labri.fr/perso/magoni/> (D. Magoni).

provides an upper bound on the length of every valid shortest path in a free multilayer network. Section 9 concludes the paper.

2. Related work

2.1. Routing algebras

Routing algebra was initially studied by Carré (1971). The author proposed an algebraic structure based on semirings for the formulation of the path computation problem. He also proposed generic solutions that were based on classical methods of linear algebra such as the Jacobi and Gauss–Seidel method, and the Gauss and Jordan elimination method. Several algebraic structures and algorithms have been described in Carré (1979), Heibisch and Weinert (1998), Gondran and Minoux (1995), Minoux (2001) to solve various optimization and routing problems in networks. Mohri (2002) proposed a new algebraic frameworks and generic algorithms for the shortest path problem and the k -shortest paths problem.

More recently, work on routing algebras has been applied to existing routing protocols. A first application of this was on path vector routing protocols proposed by Sobrinho (2003, 2005). Not only did the authors propose a sufficient condition to guarantee the correctness of the iBGP protocol, they also showed that if the algebra is *monotonic*¹ then the protocol converges in any network, but not necessarily to a global optimal solution. If the algebra is *isotonic*, however, then the protocol converges to a global optimal solution.

An approach called metarouting proposed by Griffin and Sobrinho (2005a,b) which is based on Sobrinho’s algebra and used to define routing protocols in a declarative and high-level way. They also introduced a “scoped” product to model the combined metric of iBGP and eBGP. The lexicographic product to combine several QoS metrics into a single composite metric proposed by Griffin et al. (Griffin and Sobrinho, 2005a; Gurney and Griffin, 2007; Dyerowicz and Griffin, 2013). They also showed which properties are required to guarantee optimal global and local solutions in the case of lexicographic products. Another so-called functional product was used by Khayou and Sarakbi (2017) to model EIGRP non-lexical metrics.

An important work proposed by Daggitt et al. (2018) showed that Sobrinho’s conditions are sufficient for a routing protocol to converge to a single solution, particularly in the case of the asynchronous Bellman-Ford algorithm. More recently, Sobrinho and Ferreira (2020) studied the problem of routing with non-isotonic metrics and proposed new routing protocols based on a partial order in order to preserve isotonicity.

Botero et al. have proposed the use of *paths algebras* to solve the link mapping (Botero et al., 2013) and node mapping (Hesselbach et al., 2016) stages of the virtual network embedding problem. In this case, virtual networks, quite similar to multilayer networks in essence, are constructed beforehand as opposed to the automatic and dynamic setup of routing paths studied here.

2.2. Routing algorithms with automatic tunneling

The first routing algorithm with automatic tunneling presented by Kuipers and Dijkstra (2009), which explored all possible paths using a breadth-first search and gave an exponential worst-case complexity. The related problem was shown to be NP-hard under bandwidth constraints. Later, Lamali et al. (2013), proposed a polynomial algorithm based on a language theory approach. This approach was able to find the best path between two nodes without bandwidth constraints by

¹ Ferreira (Sobrinho and Ferreira, 2020) uses the term *inflation*, while Daggitt (Daggitt et al., 2018) uses the term *increasing* to refer to the monotonicity property defined by Sobrinho (2005). In this paper, we use the term *monotonicity*.

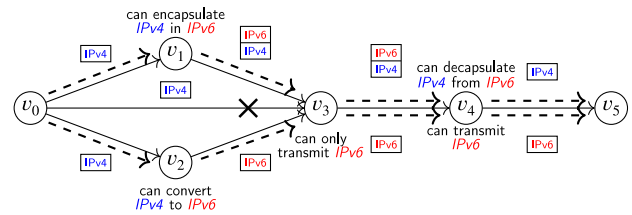


Fig. 1. Example of a network encompassing IPv4 and IPv6 protocol conversions and encapsulations.

minimizing either the number of hops or the number of adaptation functions. A matrix model for finding shortest paths with a length of at most k was proposed by Iqbal et al. (2015). However the related algorithms had an exponential complexity and could not compute paths containing cycles.

In 2016, an algorithm for shortest path computation based on any additive metric chosen by the user (e.g., weighted links, weighted adaptation functions, etc.) was proposed by Lamali et al. (2016) as well as heuristics to compute them under bandwidth constraints. This paper also described an exponential algorithm yielding the valid shortest path under various QoS constraints by generalizing the work of Van Mieghem and Kuipers (2004). The downside to these algorithms is that they are not distributed and do not build routing tables, only paths. Source routing could, nonetheless, leverage them.

The first distributed algorithm based on a generalization of the Bellman-Ford algorithm was then proposed by Lamali et al. (2019). This algorithm propagates the protocol stack in addition to the distance. A maximum protocol stack height is enforced to ensure termination in the face of cycles. This algorithm is able to construct routing tables and is therefore suitable for hop-by-hop routing. The path computation process automatically determines tunnel endpoints in valid paths and installs them in the routing tables. Another algorithm for routing with automatic tunneling was proposed by Mouhoub et al. (2022b,a). The main idea of this algorithm is to construct routing tables by using a transitive closure operation. It is a generalization of the Floyd–Warshall algorithm in which the path concatenation depends on the compatibility of protocol stacks. As it requires shared memory, this algorithm can be used inside an SDN controller, which will then propagate the resulting routing tables to the SDN switches.

2.3. Routing algebras with tunnels

Existing work on routing algebras and their application to network routing algorithms focuses on single forwarding protocols and path finding. In reality, the network is made up of heterogeneous forwarding protocols (e.g., IPv6 and IPv4). Therefore, more research is required to support the application of these algebras to multiprotocol routing and automatic tunneling in modern networks. We aim to close this gap by studying the application of routing algebras in a multilayer network model, where tunnels are omnipresent.

An example of such a network where IPv4 and IPv6 protocols coexist is shown in Fig. 1. In the (top) path between nodes v_0 and v_5 , node v_1 encapsulates an IPv4 packet within an IPv6 packet, thus inducing a protocol stack, i.e., the stack of headers (IPv4.IPv6) (from the bottom to the top of the stack). The node v_4 performs the reverse operation by decapsulating the inner IPv4 packet from the outer IPv6 one. The subpath from v_1 to v_4 is a tunnel. The whole path allows the communication from v_0 to v_5 . We call such a path a *valid path*. Another valid path is the (bottom) path, node v_2 converts an IPv4 packet into an IPv6 one. The node v_4 performs its second operation by transmitting, without any change, the IPv6 packet to the node v_5 . In the directed link from v_0 to v_3 , the node v_3 is not, however, able to transmit an IPv4 packet as is, v_5 cannot, thus, be reached from v_0 . We call such a path an *invalid path*.

As mentioned in the previous section, the only known algorithm for routing with automatic tunneling is the stack-vector algorithm proposed in Lamali et al. (2019). Based on this algorithm, our main idea is to define new algebraic structures based on: a semiring as described in Carré (1971), Mohri (2002), Dyerowicz and Griffin (2013), Sobrinho's algebra as proposed in Sobrinho (2003, 2005), Griffin and Sobrinho (2005a) and an algebra of functions² as proposed in Daggitt and Griffin (2018), Daggitt et al. (2018). A quadrant model was proposed by Gurney and Griffin (2007) to summarize the different existing styles and their relations. These new structures must have the capacity to enumerate all sets of valid shortest paths between every pair of nodes in the network. These paths can be represented by pairs of protocol stacks (or a compositions of adaptation functions that make a path as valid) and weights.

In order to define these new structures, we aim to construct two operations on the *valid shortest paths* set. The first operation is used to concatenate valid shortest paths and the second one is used to choose between them. The concatenation operation must make it possible to calculate the weight sum of a path under the condition of preserving its validity. However, the second operation must keep all valid paths. In equality cases, i.e., valid paths having the same protocol stack (tunnel), the second operation must keep the shortest one.

These proposed algebraic structures can be used to prove some convergence properties of the stack-vector protocol proposed by Lamali et al. (2019). They can also be used to find the optimal solution for networks running ECMP when using QoS metrics as shown by Dyerowicz and Griffin (2013).

2.4. Real-world multilayer networks

In real-world deployments, multilayer networks can be found as IPv4/IPv6 networks (Raste and Kulkarni, 2008), as IP/MPLS/optical networks (Urta et al., 2007; Ricciardi et al., 2015; Katib and Medhi, 2011; Amorim and Pavani, 2021; Xingfu and Deqiang, 2021), as IP/EON (Etezadi et al., 2022), as GMPLS IP/optical networks (Halabi et al., 2011; Harhira and Pierre, 2008), and in satellite networks (Guo et al., 2013). Information-centric networking (ICN) over WDM networks have also been proposed by AL-Naday et al. (2017). Multilayer networks can also be found in Industrial IoT (Wu et al., 2021). They also arise when using protocol conversions between private and public addressing spaces (Tseng et al., 2013).

3. Algebraic models

As mentioned in the previous section, there are three styles of routing algebra: semiring as described by Carré (1971), Mohri (2002), Dyerowicz and Griffin (2013), Sobrinho algebra as proposed by Sobrinho (2003, 2005), Griffin and Sobrinho (2005a) and an algebra of functions as proposed by Daggitt and Griffin (2018), Daggitt et al. (2018). A quadrant model was proposed by Gurney and Griffin (2007) to summarize the different existing styles and their relationships. In this section, we define the different styles of routing algebra with examples of traditional routing problems. The basic algebraic axioms used in this paper are defined in the Table 1. All acronyms for algebraic structures used in this paper are listed in Table 5 of the Appendix

3.1. Semiring

A set S of attributes represents a flexible measure for arbitrary routing metrics. Each link and path in a network is linked to an

² An algebra of functions is a generalization of an algebra of monoid endomorphisms, see Minoux and Gondran (2008) for more information.

Table 1
Algebraic properties.

Property	Definition
Selectivity (\oplus)	$a \oplus b \in \{a, b\}$
Idempotence (\oplus)	$a \oplus a = a$
Commutativity (\oplus)	$a \oplus b = b \oplus a$
Associativity (\oplus)	$a \oplus (b \oplus c) = (a \oplus b) \oplus c$
Left-Distributivity (\oplus, \otimes)	$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
Right-Distributivity (\oplus, \otimes)	$(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$
Identity ($\oplus, \bar{0}$)	$a \oplus \bar{0} = \bar{0} \oplus a = a$
Annihilator ($\oplus, \bar{1}$)	$a \oplus \bar{1} = \bar{1} \oplus a = \bar{1}$

attribute, determined by a *binary extension operation* denoted as \otimes . This operation is assumed to be both associative and commutative, featuring an identity attribute $\bar{1}$.

Let $a_{i,j}$ represent the attribute of link (v_i, v_j) . For a given path $p = v_0, v_1, v_2, \dots, v_{k-1}, v_k$, the attribute $a(p)$ is the result of combining the attributes of its links using the binary extension operation:

$$a(p) = a_{0,1} \otimes a_{1,2} \otimes \dots \otimes a_{k-1,k} \quad (1)$$

The attribute of a trivial path, containing just one node, is $\bar{1}$. An optimal attribute of a path from node v_i to node v_j , denoted by $a(p)^*$ is the most preferred attribute among all path attributes from node v_i to node v_j . It is obtained through a commutative *binary selection operation*, denoted by \oplus . If p_1, p_2, \dots, p_ℓ represent all paths from v_i to v_j , the optimal attribute of the optimal path p from v_i to v_j is calculated as follows:

$$a(p)^* = a(p_1) \oplus a(p_2) \oplus \dots \oplus a(p_\ell) \quad (2)$$

The null attribute $\bar{0}$ is considered the least preferred of all attributes and signifies the absence of a valid path. Using an idempotent operation \oplus , we can define a *partial order relation* \leq_{\oplus} on the attributes of S , which is reflexive, antisymmetric, and transitive. For all $a, b, c \in S$, reflexivity means that $a \leq_{\oplus} a$, antisymmetry means that $a \leq_{\oplus} b$ and $b \leq_{\oplus} a$ implies $a = b$, and transitivity means that $a \leq_{\oplus} b$ and $b \leq_{\oplus} c$ implies $a \leq_{\oplus} c$. We express this as:

$$(a \leq_{\oplus} b) \equiv (a = a \oplus b) \quad (3)$$

$$(a <_{\oplus} b) \equiv (a = a \oplus b) \text{ and } (a \neq b) \quad (4)$$

Note that when the operation \oplus is selective, this relation becomes a *total order* with $a \leq_{\oplus} b$ or $b \leq_{\oplus} a$ for all $a, b \in S$.

According to this context, the generalized routing problem will be captured in structure, called *semiring* (SM) of the form $(S, \oplus, \otimes, \bar{0}, \bar{1})$. For standard shortest path routing, the semiring SM_{sp} takes the form $(\mathbb{R}_+^{\infty}, \min, +, \infty, 0)$ with the order relation \leq defined on the set \mathbb{R}_+^{∞} . Note that \mathbb{R}_+^{∞} is the set of natural numbers $\mathbb{R}_+ \cup \{\infty\}$.

Now, the optimal solution of the generalized routing problem is formalized using the semiring structure. Let $\mathbb{M}_n(S)$ be the set of $n \times n$ matrices over S . The sum and the product of two matrices \mathbf{X} and \mathbf{Y} are defined as:

$$(\mathbf{X} \oplus \mathbf{Y})_{i,j} = \mathbf{X}_{i,j} \oplus \mathbf{Y}_{i,j} \quad (5)$$

$$(\mathbf{X} \otimes \mathbf{Y})_{i,j} = \bigoplus_{k=1}^n \mathbf{X}_{i,k} \otimes \mathbf{Y}_{k,j} \quad (6)$$

Given a network of n nodes with attributed edges, we represent the topology by an $n \times n$ adjacency matrix $\mathbf{A} \in \mathbb{M}_n(S)$ where $\mathbf{A}_{i,j} \in S$ is the attribute $a_{i,j}$ of the link from node v_i to node v_j . Missing links are represented by the null attribute $\bar{0}$. The recursive definition of the power of the matrix \mathbf{A} is given by:

$$\mathbf{A}^0 = \mathbf{A}^{(0)} = \mathbf{I} \quad (7)$$

$$\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A}^{k-1} \quad (8)$$

$$\mathbf{A}^{(k)} = \mathbf{A}^k \oplus \mathbf{A}^{(k-1)} \quad (9)$$

Here, \mathbf{I} is the identity matrix, where $\mathbf{I}_{i,j}$ is the trivial attribute $\bar{1}$ if the nodes v_i and v_j are the same, and the null attribute $\bar{0}$ otherwise.

The *global optimal solution* for the generalized routing problem consists in finding (if it exists) the \mathbf{A}^* matrix,

$$\mathbf{A}^* = \bigoplus_{p \in \mathcal{P}} a(p) \quad (10)$$

Where $\mathcal{P}_{i,j}$ is the set of all possible paths from node v_i to node v_j . In the case of the shortest paths problem non-negative weights, the global optimal solution \mathbf{A}^* converges to the matrix $\mathbf{A}^{(n-1)}$.

3.2. Algebra of functions

A potential extension of the semiring concept is the *algebra of functions* (AF), wherein the traditional semiring operation \otimes is generalized to a set of functions acting on the attribute set S . Unlike the rigid structure of $a \otimes b$, this allows for more versatile transformations of attributes, opening up the possibility for a broader range of computations. Instead of confining attribute computations to $a \otimes b$, we can now consider a more diverse set of transformations or even incorporate any function defined on S . The functions are assumed to be distributive over the \oplus operation, adhering to the property $f(a \oplus b) = f(a) \oplus f(b)$.

For every semiring $(S, \oplus, \otimes, \bar{0}, \bar{1})$, there is a corresponding algebra of functions $(S, \oplus, F, \bar{0}, \bar{1})$, where F_\otimes is the set $\{f_a(b) = a \otimes b \mid a \in S\}$ for an arbitrary operation \otimes . The operation \oplus is assumed to be selective. The null attribute $\bar{0}$ is required to be a fixed point for all function with $f(\bar{0}) = \bar{0}$. The trivial attribute $\bar{1}$ is required to be an annihilator of the operation \oplus . For standard shortest path routing, the algebra of functions shortest paths AF_{sp} takes the form $(\mathbb{R}_+^\infty, \min, F_+, \infty, 0)$.

In order to formalize the optimal solution of the generalized routing problem using an algebra of functions, we represent the topology by an $n \times n$ adjacency matrix \mathbf{A} where $\mathbf{A}_{i,j} \in F$ is the function $f_{i,j}$ of the link from node v_i to node v_j . Missing links are represented by the constant function $f(a) = \bar{0}$. We define the global computation by a matrix $\mathbf{X} \in \mathbb{M}_n(S)$ where $\mathbf{X}_{i,j}$ is the optimal attribute from node v_i to node v_j . We now define the application of \mathbf{A} on \mathbf{X} with:

$$\mathbf{A}(\mathbf{X})_{i,j} = \bigoplus_{k=1}^n \mathbf{A}_{i,k}(\mathbf{X}_{k,j}) \quad (11)$$

The k -application of matrix \mathbf{A} on \mathbf{X} is recursively given by:

$$\mathbf{X}^0 = \mathbf{I} \quad (12)$$

$$\mathbf{X}^k = \mathbf{A}(\mathbf{X}^{k-1}) \quad (13)$$

3.3. Sobrinho algebra

An algebra of functions $(S, \oplus, F, \bar{0}, \bar{1})$ equipped with an order relation \leq_\oplus over the attribute set S is termed an *ordered algebra of functions* and takes the form $(S, \leq_\oplus, F, \bar{0}, \bar{1})$. This style of algebra include a special case called Sobrinho's algebra (SA) as defined in [Griffin and Sobrinho \(2005a,b\)](#) and represented as $(S, \leq_\oplus, L, \otimes)$. Here, S denotes a set of attributes referred to as *signatures*, \leq_\oplus is total order over signatures, L stands for a set of *labels*, and \otimes is a binary operation that associates for a pair (signature, label) a signature. As an ordered algebra of functions, Sobrinho's algebra is expressed as (S, \leq_\oplus, F_L) , where $F_L = \{f_\ell(a) = \ell \otimes a \mid \ell \in L \text{ and } a \in S\}$. Hence, we can consider the pair (L, \otimes) as a mechanism for indexing the set of functions F_L .

In this paper we adopte the original notation ([Sobrinho, 2005](#)) that slightly deviates from the one presented above. Following the original format $(W, \leq, L, S, \bar{0}, \otimes, \omega)$, the set of signatures S is accompanied by an associated set of totally ordered weights W with \leq and a weight function ω maps S into W . The signature $\bar{0}$ assumes a special role within S and is linked to invalid paths. For every signature $a \in S - \{\bar{0}\}$, it holds $\omega(a) \leq \omega(\bar{0})$. Notably, the extension of an invalid path remains an invalid path, as for all $l \in L$, $l \otimes \bar{0} = \bar{0}$. With this formulation,

Table 2
Network model notations.

Notation	Description
\mathcal{N}	Multilayer network
\mathcal{A}	The set of all communication protocols
\mathcal{G}	A directed graph
\mathcal{V}	The set of all nodes
\mathcal{E}	The set of all links
\mathcal{H}	The set of protocol stacks
H	A set of protocol stacks in \mathcal{H}
\mathcal{F}	The set of all elementary adaptation functions
\mathcal{F}_{id}	The set of all passive adaptation functions
\mathcal{F}	A set of adaptation functions in \mathcal{F}
$\hat{\mathcal{F}}$	The set of all adaptation functions closed under composition
\hat{F}	A set of adaptation functions in $\hat{\mathcal{F}}$
\odot	The composition operation of adaptation functions
\diamond	The concatenation operation of protocol stacks

the shortest paths routing can be modeled by the Sobrinho's algebra, $SA_{sp} = (\mathbb{R}_+^\infty, \leq, \mathbb{R}_+, \mathbb{R}_+^\infty, \infty, +, id_{\mathbb{R}_+^\infty})$.

In order to formalize the optimal solution of the generalized routing problem using an algebra of functions, we represent the topology by an $n \times n$ adjacency matrix \mathbf{A} where $\mathbf{A}_{i,j} \in L$ is the label $\ell_{i,j}$ of the link from node v_i to node v_j . Missing links are represented by the special signature $\bar{0}$.

For a given global computation matrix $\mathbf{X} \in \mathbb{M}_n(S)$ where $\mathbf{X}_{i,j}$ is the optimal signature from node v_i to node v_j . We define the product of \mathbf{A} and \mathbf{X} recursively as follows:

$$\mathbf{X}^0 = \mathbf{I} \quad (14)$$

$$\mathbf{X}^k = \mathbf{X}^{k-1} \otimes \mathbf{A} \quad (15)$$

4. Multilayer network model

We use the same network model and definitions as those presented in [Lamali et al. \(2018, 2019\)](#) along with some additional definitions. [Table 2](#) summarizes the main notations of the paper.

4.1. Multilayer network

Definition 1. A multilayer network is modeled by 4-tuple $\mathcal{N} = (\mathcal{G}, \mathcal{A}, \mathcal{F}, \omega)$ where:

- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph modeling the network topology. The set of nodes \mathcal{V} models the routers of the network. The set of links \mathcal{E} models the physical links between the routers. The number of nodes is denoted by $|\mathcal{V}| = n$ and the number of links is denoted by $|\mathcal{E}| = m$.
- $\mathcal{A} = \{x, y, z, \dots\}$ is the set of protocols available in the network, but not necessarily at each router. The number of protocols is denoted by $|\mathcal{A}| = \lambda$. The set of protocols that a node v can receive is denoted by $In(v)$, and the set of protocols that v can send is denoted by $Out(v)$.
- \mathcal{F} is the set of elementary adaptation functions available in the network. For each node $v \in \mathcal{V}$, $\mathcal{F}(v)$ is the set of adaptation functions available on node v . There are three types of adaptation functions:

- Conversion:** the header of a packet of protocol x is transformed into the header of another protocol y . It is denoted by $(x \rightarrow y)$. If the received and emitted packets are of the same protocol, *i.e.*, a classical transmission, it is denoted by $(x \rightarrow x)$. Thus, we consider a classical transmission as a special type of conversion;

Table 3

Algebraic structures for the *valid shortest paths* problem.

Algebra Style	Over Adaptation Functions	Over Protocol Stacks
Semiring (SR_{vsp})	$(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty}), \cup, (\otimes \times +), \emptyset, (F_{id} \times 0))$	$(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \cup, (\otimes \times +), \emptyset, (\{(\epsilon, \epsilon)\} \times 0))$
Algebra of Functions (AF_{vsp})	$(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty}), \cup, \mathcal{P}(F_{\otimes} \times F_+), \emptyset, (\hat{\mathcal{F}} \times 0))$	$(\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty}), \cup, \mathcal{P}(F \times F_+), \emptyset, (\mathcal{H} \times 0))$
	–	$(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \cup, \mathcal{P}(F_{\otimes} \times F_+), \emptyset, (\mathcal{H}^2 \times 0))$
	$(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(F \times \mathbb{R}_+), \mathcal{P}(F \times \mathbb{R}_+^{\infty}), \emptyset, (\otimes \times +), id_{\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty})})$	$(\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(F \times \mathbb{R}_+), \mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty}), \emptyset, (\otimes \times +), id_{\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty})})$
Sobrinho's Algebra (SA_{vsp})	$(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(F), \mathcal{P}(\hat{\mathcal{F}}), \emptyset, \otimes, \omega)$	$(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(F), \mathcal{P}(\mathcal{H}), \emptyset, \otimes, \omega)$
	–	$(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+), \mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \emptyset, (\otimes \times +), id_{\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty})})$
	–	$(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(\mathcal{H}^2), \mathcal{P}(\mathcal{H}^2), \emptyset, \otimes, \omega)$

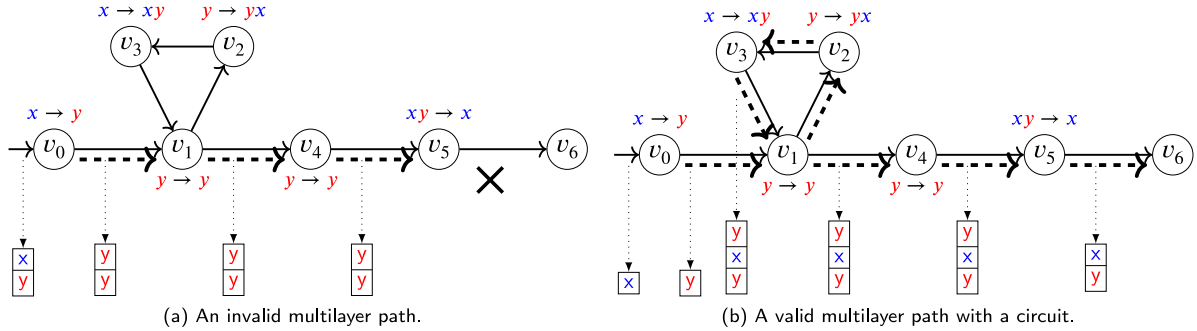


Fig. 2. Example of an invalid path and a valid path with a circuit.

- **Encapsulation:** the whole packet of protocol x is encapsulated in the data field of a packet of protocol y . It is denoted by $(x \rightarrow xy)$. Note that a packet can be encapsulated in another one of the same protocol, such as for IP-in-IP (i.e., RFC 1853 and RFC 2003). In this case, the function is simply denoted by $(x \rightarrow xx)$;
- **Decapsulation:** a packet of protocol x is extracted from the data field of a packet of protocol y , this is the reverse operation of $(x \rightarrow xy)$, and thus it is denoted by $(xy \rightarrow x)$. Note that this operation can be performed only if the received packet of protocol y effectively contains a packet of protocol x in its data field.

- $\omega : \mathcal{V} \times \mathcal{F} \times \mathcal{V} \rightarrow \mathbb{R}_+$ is a weight function modeling any additive metric. The value of $\omega(v_i, f, v_j)$ is the weight of performing the adaptation function f on v_i and then transmitting the packet on the link (v_i, v_j) . Its sum over a path is the cost that we want to minimize. It is very generic as it can model the number of hops (by putting $\omega(v_i, f, v_j) = 1$ for all the possible triples), or the number of encapsulations (by putting $\omega(v_i, f, v_j) = 1$ when f is an encapsulation and $\omega(v_i, f, v_j) = 0$ otherwise), etc.

4.2. Protocol stack

A sequence of adaptation functions induces a protocol stack. For example, the sequence:

$$(x \rightarrow y)(y \rightarrow yx)(x \rightarrow x)(x \rightarrow xx)(x \rightarrow xy)$$

induces the stack $yxxxy$ (top on the right). The protocol at the top of the stack h is denoted by $Top(h)$.

Let f be an adaptation function and h a protocol stack. We denote by $f(h)$ the application of the adaptation function f to the stack h . For example, if f is the decapsulation $(xy \rightarrow x)$ and $h = yxy$, then $f(h) = yx$. The outer protocol y is removed and the inner one (protocol x) is extracted, without any change to the underlying protocols that are still nested (here the bottom y).

Applying an adaptation function to a protocol stack is not always possible. For example, it is impossible to apply $f = (yx \rightarrow y)$ to stack $h = x$ since there is no nested (packet of) protocol y in h . In such cases, we denote the resulting stack by $f(h) = \phi$, where ϕ is called the *forbidden stack*. For any adaptation function f , we have $f(\phi) = \phi$. This means that there is an impossibility to apply an adaptation function at some point of a path and thus there is no way to continue through this path.

Definition 2. Let $f_i f_{i+1} \dots f_{j-1} f_j$ a sequence of adaptation functions and h_i an initial protocol stack. We define recursively the induced protocol stack h_k by,

$$h_k = f_{k-1}(h_{k-1}), \quad i + 1 \leq k \leq j + 1 \tag{16}$$

4.3. Valid path

A *multilayer path* is represented by a directed path in the underlined graph \mathcal{G} by taking into account the adaptation function involved at each node in the path. Thus, a path from a node v_i to a node v_j in \mathcal{N} is a mixed sequence of nodes and adaptation functions with starting and ending protocol stacks $h_i v_i f_i v_{i+1} f_{i+1} \dots v_{j-1} f_{j-1} h_j v_j$ where each v_k , $i \leq k \leq j$ is a node, and each f_k is an adaptation function. The starting stack is h_i and the ending stack is h_j .

Definition 3. A path $p = h_i v_i f_i v_{i+1} f_{i+1} \dots v_{j-1} f_{j-1} h_j v_j$ from v_i to v_j is *valid* if and only if:

- The sequence $v_i v_{i+1} \dots v_{j-1} v_j$ is a classical path in \mathcal{G} and each $f_k \in \mathcal{F}(v_k)$;
- $h_i \neq \phi$ and $Top(h_i) \in In(v_i)$;
- $h_j \neq \phi$ and $Top(h_j) \in In(v_j)$;
- $h_j = f_{j-1}(\dots f_{i+1}(f_i(h_i)))$.

In **Fig. 2(a)**, the depicted path from v_0 to v_6 with the starting protocol stack yx is:

$$v_0(x \rightarrow y)v_1(y \rightarrow y)v_4(y \rightarrow y)v_5(xy \rightarrow x)v_6$$

It cannot be valid because, the node v_5 receives the protocol stack yy and it cannot decapsulate the protocol x from y , i.e., it cannot perform the function $xy \rightarrow x$ on the stack yy . In contrast, Fig. 2(b) depicts a valid path v_0 to v_6 with the starting stack x and the arrival stack yx :

$$v_0(x \rightarrow y)v_1(y \rightarrow y)v_2(y \rightarrow yx)v_3(x \rightarrow xy) \\ v_1(y \rightarrow y)v_4(y \rightarrow y)v_5(xy \rightarrow x)v_6$$

The used links in the path are in bold and the corresponding protocol stacks are below the links in Fig. 2.

The weight of a valid path³ from node v_i to node v_j with stacks h_i and h_j , $p = h_i v_i f_i v_{i+1} f_{i+1} \dots v_{j-1} f_{j-1} h_j v_j$ is the sum of the weights of its links and its adaptation functions. It is denoted by:

$$\omega(p) = \sum_{k=i}^{j-1} \omega(v_k, f_i, v_{k+1}) \quad (17)$$

5. Routing algebras with tunnels

In this section, we first define the adaptation function composition operation, protocol stack concatenation, and the product of union-min operations. Next, we define three algebraic structures (semiring, Sobrinho's algebra, and algebra of functions) for the valid path problem vp . Then, we study the semi-direct product of the *valid paths* structures vp with the *shortest paths* structures sp in order to model the valid shortest paths vsp . Table 3 summarizes all *valid shortest paths* structures proposed in this paper.

5.1. Adaptation function composition

In the network model we explained that the successive application of adaptation functions on a stack of starting protocols induces a stack of arrival. This application can be represented by a composition of functions. More formally, let $\mathcal{H} = \{\phi, x, y, \dots, xx, yy, xy, yx, \dots\}$ be the set of all possible protocol stacks. Note that this set can be finite for a given multilayer network in which the number of protocols in any stack does not exceed λn^2 (the proof of this bound is given in Lamali et al. (2019)). We can define an elementary adaptation function as a function f from $H_a \rightarrow H_b$ where H_a (resp. H_b) is a set of protocol stacks in \mathcal{H} representing the domain (resp. co-domain) of f . For example, if the function is the decapsulation $xy \rightarrow x$, then $f : H_{xy} \rightarrow H_x$ where H_{xy} (resp. H_x) is the set of all protocol stacks starting with the sub protocol stack xy (resp. the protocol x). We define the application of a sequence of adaptation functions $f_i f_{i+1} \dots f_{j-1} f_j$ on a given protocol stack h by a composition of these adaptation functions.

Definition 4. Let $f : H_a \rightarrow H_b$ and $f' : H'_a \rightarrow H'_b$ be two elementary adaptation functions in \mathcal{F} . We define the composition by the new function $f'' = f' \circ f$ as follows:

$$f'' = \begin{cases} H'_a \rightarrow H'_b & \text{if } (H''_a \neq \emptyset) \wedge (H'_b \neq \emptyset) \\ \{\phi\} \rightarrow \{\phi\} & \text{otherwise} \end{cases} \quad (18)$$

Where:

$$H''_a = \{h \in H_a \mid f(h) \in H'_a\} \quad (19)$$

$$H''_b = \{f'(h) \mid h \in (H_b \cap H'_a)\} \quad (20)$$

Note that this composition \circ is associative and not commutative. The function $\{\phi\} \rightarrow \{\phi\}$ is denoted by the *forbidden function* $\phi \rightarrow \phi$. It is clear that the set of elementary adaptation functions \mathcal{F} is not closed under composition. For example, if the composition is $(x \rightarrow xy) \circ (y \rightarrow yx)$ then the composed function is $(y \rightarrow yxy)$. This new function is not in the set \mathcal{F} . For this, we define a new set $\hat{\mathcal{F}}$ of all adaptation functions closed under composition. Each composition of elementary adaptation functions can be represented by a new composed adaptation function.

The impossibility of composition is represented by the *forbidden function* $\phi \rightarrow \phi$.

Let $\hat{\mathcal{F}} = \{\phi \rightarrow \phi, x \rightarrow x, x \rightarrow y, x \rightarrow xxx, xyx \rightarrow x, \dots\}$ be the set of all adaptation functions closed under composition. We say that two functions f' and f of $\hat{\mathcal{F}}$ are equal if and only if they have the same domain and co-domain. For example, the two functions $(x \rightarrow xx)$ and $(y \rightarrow x) \circ (x \rightarrow xy)$ are equal and they can apply on protocol stacks starting with x and give a protocol stacks starting with xx . In the following text, we denote by \mathcal{F}_{id} the set of all passive functions, i.e., classical transmissions $\{x \rightarrow x, y \rightarrow y, \dots\}$ called *identity adaptation functions*.

Using this new operation of adaptation functions composition, a valid path $p = h_i v_i f_i v_{i+1} f_{i+1} \dots v_{j-1} f_{j-1} h_j v_j$ from v_i to v_j can be represented by the valid composition of the elementary adaptation functions $f_{j-1} \circ \dots \circ f_{i-1} \circ f_i$. This latter composition is defined by the function $h_i \rightarrow h_j$ of the set $\hat{\mathcal{F}}$. In Fig. 2(b), the composed function of the represented valid path is $x \rightarrow xy$. The composed function of invalid path is the function $\phi \rightarrow \phi$

5.2. Protocol stack concatenation

In Mouhoub et al. (2022b), the authors proposed a generalized Floyd-Warshall algorithm for automatic tunneling. This algorithm is based on the concatenation operation of protocol stacks' pairs.

Definition 5. Let $p = h_0 v_0 f_0 \dots h_k v_k$ and $p' = h'_0 v'_0 f'_0 \dots h'_k v'_k$ be two valid paths. We define the concatenation of p and p' by the new path $p'' = h''_0 v_0 f_0 \dots v_k f_k v'_0 f'_0 \dots h'_k v'_k$ where:

$$(h''_0, h'_k) = \begin{cases} (h_0, h'_k) & \text{if } h_k = h'_0 \\ (h.h_0, h'_k) & \text{if } h'_0 - h_k = h \\ (h_0, h.h'_k) & \text{if } h_k - h'_0 = h \\ (\phi, \phi) & \text{otherwise} \end{cases} \quad (21)$$

Note that $h.h'$ is the simple concatenation of the stack h and h' . For example, if $h = xy$ and $h' = x$ then $h.h' = xxy$.

Now we can define the binary operation \diamond that concatenates pairs of protocol stacks according to the previous conditions,

$$(h_i, h_j) \diamond (h'_i, h'_j) = (h''_i, h''_j) \quad (22)$$

Note that we can generalize this operation on sets of pairs of protocol stacks in order to compute the pair-wise sets.

5.3. Union-min operation

In order to enumerate the set of all valid shortest paths between each pair of nodes in a multilayer network, we introduce a semi-direct product over a set of weighted attributes,⁴ in which the concatenation operation is a direct product and the choice operation is based on a new binary operation that combines the *union* and the *min* operations.

Let $S = \{a, b, c, \dots\}$ be a finite set of attributes. We define the weighted set of S as the set product $T = S \times \mathbb{R}_+^\infty$. We denote $\mathcal{P}(T)$, the power set in which each subset contains a unique pair for each element. We define the *union-min* operation over subsets of weighted attributes as follows:

Definition 6. Let T_i and T_j be two subsets of $\mathcal{P}(T)$. We define the *union-min* operation of T_i and T_j as follows:

$$T_i \cup_{min} T_j = \left\{ (a, \omega_a) \mid (a, \omega_a) \in T_i \wedge \forall (b, \omega_b) \in T_j, \right. \\ \left. (a = b) \Rightarrow \omega_a = \omega_a \min \omega_b \right\} \quad (23)$$

³ The weight of an invalid path is not defined and can be set to ∞ value.

⁴ These attributes can be protocol stacks or compositions of adaptation functions. It depends on the nature of the algebraic structure.

This new operation, *i.e.*, union-min, introduces the idea of enumerating different weighted paths from a source to a destination. And in the case of equality of paths (with the same attributes), it keeps the path with the smallest weight. It is easy to check that the *union-min* operation is idempotent, associative and commutative, but not selective (it can return a new subset different from the two initial subsets). The identity element of this operation is the empty subset and the annihilator element is the subset of unique attributes with weight 0.

Definition 7. Let T_i and T_j be two subsets of $\mathcal{P}(T)$. We define the order relation as follows:

$$T_i \subseteq_{\min} T_j \equiv \forall (a, \omega_a) \in T_i \rightarrow \exists (b, \omega_b) \in T_j, \quad (24)$$

$$(a = b) \wedge (\omega_a \leq \omega_b)$$

It is clear that this order is a partial order, *i.e.*, there exists some incomparable subsets. For example, the two subsets $\{(a, 1), (b, 5)\}$ and $\{(a, 4), (b, 2)\}$ are incomparable.

In the next sections, we will use this operation to define our semi-direct product for semirings and algebras of functions. In the case of Sobrinho's algebra, the union-min operation is captured by the order relation.

5.4. Semiring with tunnels

5.4.1. Semiring over adaptation functions

Recall that a valid multilayer path is represented by a valid composition of adaptation functions. In order to compute the set of all valid paths, *i.e.*, valid compositions, we extend the definition of adaptation function composition on sets of compositions.

Let be $\hat{\mathcal{F}}$ the set of all adaptation functions closed under composition, and $\mathcal{P}(\hat{\mathcal{F}})$ its power set. If \hat{F}_1 and \hat{F}_2 are two subsets of $\mathcal{P}(\hat{\mathcal{F}})$, then we define the set of pair-wise compositions,

$$\hat{F}_1 \odot \hat{F}_2 = \{\hat{f}_1 \circ \hat{f}_2 \mid \hat{f}_1 \in \hat{F}_1 \text{ and } \hat{f}_2 \in \hat{F}_2\} \quad (25)$$

Note that in the case of invalid composition, the resulting forbidden function will be removed from the result set. The composition with an empty set is always an empty set. Based on this operation, we define our semiring valid paths SR_{vsp} that enumerates all valid paths as follows,

$$SR_{vsp} = (\mathcal{P}(\hat{\mathcal{F}}), \cup, \odot, \emptyset, \mathcal{F}_{id}) \quad (26)$$

Where, \emptyset is the empty set composition and \mathcal{F}_{id} is the set of identity adaptation functions $\{x \rightarrow x, y \rightarrow y, \dots\}$. It is easy to check that the composition \odot is associative and non-commutative with the \emptyset as annihilator. We check the identity set compositions \mathcal{F}_{id} and the distributivity of \odot over \cup ,

Identity of \odot

Proposition 1. The set \mathcal{F}_{id} is the identity element of the composition \odot .

Proof. Let $\hat{F} = \{\hat{f}_1, \hat{f}_{i+1}, \dots, \hat{f}_{j-1}, \hat{f}_j\}$ be a set of composed functions in $\mathcal{P}(\hat{\mathcal{F}})$. We want to show that,

$$\hat{F} \odot \mathcal{F}_{id} = \mathcal{F}_{id} \odot \hat{F} = \hat{F} \quad \forall \hat{F} \in \mathcal{P}(\hat{\mathcal{F}}) \quad (27)$$

Let $\hat{f}_k, i \leq k \leq j, \hat{f}_k : H_a \rightarrow H_b$ be a function in \hat{F} . There are two possible situations:

1. H_a is a set of protocol stacks starting with x . In this case, $\hat{f}_k \circ (x \rightarrow x) = \hat{f}_k$ and for all $f \in \mathcal{F}_{id} - \{(x \rightarrow x)\}$ we have $\hat{f}_k \circ f = (\phi \rightarrow \phi)$.
2. H_b is a set of protocol stacks starting with x . In this case, $(x \rightarrow x) \circ \hat{f}_k = \hat{f}_k$ and for all $f \in \mathcal{F}_{id} - \{(x \rightarrow x)\}$ we have $f \circ \hat{f}_k = (\phi \rightarrow \phi)$.

Note that, by the definition of \odot , the forbidden functions are removed from the result set. \square

Distributivity of \odot over \cup

Proposition 2. The composition \odot is distributive on the union \cup .

Proof. Let $\hat{F}_1, \hat{F}_2, \hat{F}_3$ be three sets of composed functions in $\mathcal{P}(\hat{\mathcal{F}})$. By the definition of \odot over sets of composed functions, which compute the set of pair-wise compositions, we can see that,

$$\hat{F}_1 \odot (\hat{F}_2 \cup \hat{F}_3) = (\hat{F}_1 \odot \hat{F}_2) \cup (\hat{F}_1 \odot \hat{F}_3) \quad (28)$$

And,

$$(\hat{F}_2 \cup \hat{F}_3) \odot \hat{F}_1 = (\hat{F}_2 \odot \hat{F}_1) \cup (\hat{F}_3 \odot \hat{F}_1) \quad \square \quad (29)$$

Now, we can define the *valid shortest paths* semiring SR_{vsp} by the following semi-direct product of SR_{vp} and SR_{sp} ,

$$SR_{vsp} = \left(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^\infty), \cup_{\min}, (\odot \times +), \emptyset, (\mathcal{F}_{id} \times 0) \right) \quad (30)$$

It is well known that the direct product of semigroups preserves the associativity property, the identity attributes and the annihilators, as indicated in [Harden \(1949\)](#). In most situations where algebraic structures violate some axioms of semirings, they do not generally satisfy the distributivity of \otimes over \oplus . We check this property in order to ensure that our structure defines a semiring.

Distributivity of $(\odot \times +)$ over \cup_{\min}

Proposition 3. The direct product $(\odot \times +)$ is distributive on the union-min product \cup_{\min} .

Proof. We want to show that for all $\hat{F}_1, \hat{F}_2, \hat{F}_3$ sets of weighted composed functions in $\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^\infty)$ we have $lhs = rhs$ where,

$$lhs = \hat{F}_1 (\odot \times +) (\hat{F}_2 \cup_{\min} \hat{F}_3) \quad (31)$$

$$rhs = (\hat{F}_1 (\odot \times +) \hat{F}_2) \cup_{\min} (\hat{F}_1 (\odot \times +) \hat{F}_3) \quad (32)$$

Based on the definition of the union-min operation, we distinguish between the two following cases:

Case 1: The two sets \hat{F}_2 and \hat{F}_3 are strictly different and do not have any common composed function,

$$\hat{F}_2 \cup_{\min} \hat{F}_3 = \hat{F}_2 \cup \hat{F}_3$$

In this case, we can see that,

$$lhs = rhs = (\hat{F}_1 (\odot \times +) \hat{F}_2) \cup (\hat{F}_1 (\odot \times +) \hat{F}_3) \quad (33)$$

Case 2: The two sets have some common composed functions,

$$\hat{F}_2 \cup_{\min} \hat{F}_3 = \hat{F}_2^* \cup \hat{F}_3^* \quad (34)$$

Where: $\hat{F}_2 = \hat{F}_2^* \cup \hat{F}$ and $\hat{F}_3 = \hat{F}_3^* \cup \hat{F}$ and \hat{F} (resp. \hat{F}') is the non empty set of the common non optimal composed functions of \hat{F}_2 (resp. \hat{F}_3). In this situation, we have,

$$\hat{F}_1 (\odot \times +) \hat{F}_2 = (\hat{F}_1 (\odot \times +) \hat{F}_2^*) \cup (\hat{F}_1 (\odot \times +) \hat{F}) \quad (35)$$

$$\hat{F}_1 (\odot \times +) \hat{F}_3 = (\hat{F}_1 (\odot \times +) \hat{F}_3^*) \cup (\hat{F}_1 (\odot \times +) \hat{F}) \quad (36)$$

And,

$$(\hat{F}_1 (\odot \times +) \hat{F}_2^*) \cup_{\min} (\hat{F}_1 (\odot \times +) \hat{F}') = (\hat{F}_1 (\odot \times +) \hat{F}_2^*) \quad (37)$$

$$(\hat{F}_1 (\odot \times +) \hat{F}_3^*) \cup_{\min} (\hat{F}_1 (\odot \times +) \hat{F}) = (\hat{F}_1 (\odot \times +) \hat{F}_3^*) \quad (38)$$

As we can see, the two sets \hat{F}_2^* and \hat{F}_3^* are different. So we can apply the case 1,

$$lhs = rhs = (\hat{F}_1 (\odot \times +) \hat{F}_2^*) \cup (\hat{F}_1 (\odot \times +) \hat{F}_3^*) \quad (39)$$

We can check the right distributivity in the same way. \square

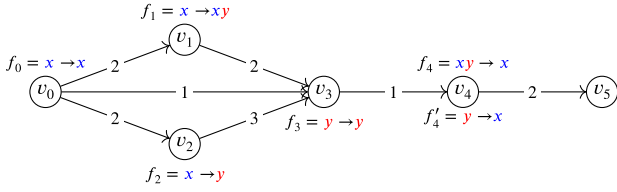


Fig. 3. Simple example network for valid shortest paths.

Using this semiring, we can define the adjacency matrix \mathbf{A} of all valid paths where $\mathbf{A}_{i,j}$ is the set of valid paths from node i to node j represented by the corresponding compositions of adaptation functions and their weights involved in each path. Missing paths are simply represented by the empty set of compositions.

In order to explain how it works, we take the same network of Fig. 1 by adding weights on links. We also add the function $(x \rightarrow x)$ to model that node v_0 can emit packets of protocol x . The function $(y \rightarrow y)$ will be replaced by the conversion $(y \rightarrow x)$ in order to create two equal compositions between nodes v_0 and v_5 . These modification are presented in Fig. 3.

We now define the adjacency matrix \mathbf{A} of the network presented in Fig. 3,

$$\mathbf{A} = \begin{matrix} & \begin{matrix} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \left[\begin{array}{cccccc} \emptyset & \{(f_0, 2)\} & \{(f_0, 2)\} & \{(f_0, 1)\} & \emptyset & \emptyset \\ & \emptyset & \emptyset & \{(f_1, 2)\} & \emptyset & \emptyset \\ & & \emptyset & \{(f_2, 3)\} & \emptyset & \emptyset \\ & & & \emptyset & \{(f_3, 1)\} & \emptyset \\ & & & & \emptyset & \{(f_4, 2), (f_4', 2)\} \\ & & & & & \emptyset \end{array} \right] \end{matrix}$$

In this example, we are interested in computing the set of valid shortest paths, i.e., the valid shortest compositions from node v_0 to node v_5 . Note that the intermediate paths will be obtained directly by calculating the \mathbf{A}^* matrix. For this, we use the formula defined in Section 3 to compute the \mathbf{A}^* matrix. The final result of the latter is shown in Fig. 4.

5.4.2. Semiring over protocol stacks

Another way to define a semiring on the set of protocol stacks is based on the concatenation operation of pairs of protocol stacks defined in Section 5.2. However, this new operation does not have a identity attribute. One possible solution to this problem is to add an empty stack (ϵ) to the set of stacks \mathcal{H} . In this context, the empty path represented by the pair of empty stacks (ϵ, ϵ) serves as the identity attribute for stack concatenation, defined as follows,

$$(h_i, h_j) \diamond (\epsilon, \epsilon) = (\epsilon, \epsilon) \diamond (h_i, h_j) = (h_i, h_j) \quad (40)$$

Based on this, we can define a semiring SM_{vp} on the set of stacks,

$$SM_{vp} = \left(\mathcal{P}(\mathcal{H}^2), \cup, \diamond, \emptyset, \{(\epsilon, \epsilon)\} \right) \quad (41)$$

Now, we can define the semiring SM_{vsp} on the set of stacks as the product of two semirings, SM_{vp} and SM_{sp} ,

$$SM_{vsp} = \left(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^\infty), \cup, \diamond, (\diamond \times +), \emptyset, \{(\epsilon, \epsilon)\} \times 0 \right) \quad (42)$$

Note that the properties of this structure can be verified in the same manner as the previous semiring on the set of compositions.

5.5. Algebra of functions with tunnels

5.5.1. Algebra of functions over adaptation functions

In order to define an algebra of functions in which the valid shortest paths are represented by the valid shortest compositions of adaptation

functions, we can directly use the semiring SR_{vp} and get the following algebra,

$$AF_{vp} = \left(\mathcal{P}(\hat{\mathcal{F}}), \cup, \mathcal{P}(F_\odot), \emptyset, \hat{\mathcal{F}} \right) \quad (43)$$

Now, we define the valid shortest paths algebra AF_{vsp} by the semi-direct product of AF_{vp} and AF_{sp} as follows,

$$AF_{vsp} = \left(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^\infty), \cup, \mathcal{P}(F_\odot \times F_+), \emptyset, (\hat{\mathcal{F}} \times 0) \right) \quad (44)$$

Where: $F_\odot \times F_+$ is the set of functions that apply on weighted compositions defined by the product set,

$$\left\{ (\hat{f}_a, f_{\omega_a})(b, \omega_b) = (a \odot b, \omega_a + \omega_b) \mid b \in \hat{\mathcal{F}}, \omega_b \in \mathbb{R}_+^\infty \right\} \quad (45)$$

Note that applying a function $(\hat{f}_a, f_{\omega_a})$ on the empty set of weighted compositions is the empty set of weighted compositions. This defines a fixed point for all functions in the power set. The set of all compositions with weight 0 defines the annihilator set for the union-min operation.

5.5.2. Algebra of functions over protocol stacks

We define here an algebra over protocol stacks. This algebra can generalize the idea of the Bellman-Ford algorithm with the automatic tunneling proposed in Lamali et al. (2019) in the stack-vector protocol. Recall that our elementary adaptation functions are defined from $\mathcal{H} \rightarrow \mathcal{H}$, so we can define an algebra of functions on the set of all protocol stacks \mathcal{H} . For this, we define the application of a set of adaptation functions on a set of protocol stacks.

Let $F = \{f_1, \dots, f_\ell\}$ be a set of adaptation functions in $\mathcal{P}(\mathcal{F})$ and $H = \{h_1, \dots, h_k\}$ a set of protocol stacks in $\mathcal{P}(\mathcal{H})$. We define the application of F on H ,

$$F(H) = \left\{ f_i(h_j) \mid f_i \in F \text{ and } h_j \in H \right\} \quad (46)$$

In particular, if the set H or F is the empty set, then $F(H)$ is also the empty set. We define the AF_{vp} algebra,

$$AF_{vp} = \left(\mathcal{P}(\mathcal{H}), \cup, \mathcal{P}(F), \emptyset, \mathcal{H} \right) \quad (47)$$

Using the semi-direct product with the AF_{sp} we obtain,

$$AF_{vsp} = \left(\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^\infty), \cup, \mathcal{P}(F \times F_+), \emptyset, (\mathcal{H} \times 0) \right) \quad (48)$$

Where $F \times F_+$ is the set of functions that apply on weighted protocol stacks defined by the product set,

$$\left\{ (f, f_\omega)(h, \omega_h) = (f(h), \omega + \omega_h) \mid h \in \mathcal{H}, \omega_h \in \mathbb{R}_+^\infty \right\} \quad (49)$$

Note that applying a function (\hat{f}, f_ω) on the empty set of weighted protocol stacks is the empty set of weighted protocol stacks. This defines a fixed point for all functions in the power set. The set of all protocol stacks with weight ∞ defines the annihilator set for the union-min operation.

Using this algebra, we can define the adjacency matrix \mathbf{A} of adaptation functions, where $\mathbf{A}_{i,j}$ is the set of adaptation functions of the node i with the set of corresponding weight functions. Missing links are represented by the empty set of pair-wise functions. We can define the routing state by the matrix \mathbf{X} where $\mathbf{X}_{i,j}$ is the set of all possible shortest protocol stacks that allow to route from node i to j .

In Section 5.2, we have defined a binary concatenation operation of pairs of protocol stacks in order to model the transitive closure operation to compute valid paths by transitive closure. Using the previous definition, we define the following algebra,

$$AF_{vp} = \left(\mathcal{P}(\mathcal{H}^2), \cup, \mathcal{P}(F_\odot), \emptyset, \mathcal{H}^2 \right) \quad (50)$$

Using the semi-direct product with the AF_{sp} we obtain,

$$AF_{vsp} = \left(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^\infty), \cup, \mathcal{P}(F_\odot \times F_+), \emptyset, (\mathcal{H}^2 \times 0) \right) \quad (51)$$

$$A^* = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ \left[\begin{array}{cccccc} F_{id} \times 0 & \{(f_0, 2)\} & \{(f_0, 2)\} & \{(f_0, 1), (f_1 \circ f_0, 4), (f_2 \circ f_0, 5)\} & \{(f_3 \circ f_1 \circ f_0, 5), (f_3 \circ f_2 \circ f_0, 6)\} & \{(f_4 \circ f_3 \circ f_1 \circ f_0, 7), (f'_4 \circ f_3 \circ f_1 \circ f_0, 7)\} \\ & F_{id} \times 0 & \emptyset & \{(f_1, 2)\} & \{(f_3 \circ f_1, 3)\} & \{(f_4 \circ f_3 \circ f_1, 5), (f'_4 \circ f_3 \circ f_1, 5)\} \\ & & F_{id} \times 0 & \{(f_2, 3)\} & \{(f_3 \circ f_2, 4)\} & \{(f_4 \circ f_3 \circ f_2, 6), (f'_4 \circ f_3 \circ f_2, 6)\} \\ & & & F_{id} \times 0 & \{(f_3, 1)\} & \{(f_4 \circ f_3, 3), (f'_4 \circ f_3, 3)\} \\ & & & & F_{id} \times 0 & \{(f_4, 2), (f'_4, 2)\} \\ & & & & & F_{id} \times 0 \end{array} \right] \end{matrix} \begin{matrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix}$$

Fig. 4. A* matrix with all valid shortest paths of the network presented in Fig. 3.

Where: $F_{\circ} \times F_{+}$ is the set of functions that apply on weighted pairs of protocol stacks defined by the product set,

$$\{(f_h, f_{\omega})(h, \omega) = (h \circ h', \omega + \omega'_h), | h' \in \mathcal{H}^2, \omega'_h \in \mathbb{R}_+^{\infty}\} \quad (52)$$

Note that applying a function (f_h, f_{ω}) on the empty set of weighted pairs of protocol stacks yields the empty set. This defines a fixed point for all functions in the power set. The set of all pairs of protocol stacks with weight 0 defines the annihilator set for the union-min operation.

The most common violation among all of our algebras of functions is that they lack selectivity, which is related to the nature of the union-min operation.

5.6. Sobrinho's algebra with tunnels

5.6.1. Sobrinho's algebra over adaptation functions

We define the following Sobrinho's algebra for the valid paths problem,

$$SA_{vp} = \left(\mathcal{P}(\hat{\mathcal{F}}), \subseteq, \mathcal{P}(\mathcal{F}), \mathcal{P}(\hat{\mathcal{F}}), \emptyset, \odot, \text{id}_{\mathcal{P}(\hat{\mathcal{F}})} \right) \quad (53)$$

The signatures are the sets of valid compositions of adaptation functions and the labels are defined as sets of adaptation functions. The extension operation is the composition of sets of compositions. The special signature is the empty set of compositions and the weight function is the identity function of the power set of compositions. The order relation is given by the partial order \subseteq defined over subsets of compositions. Note that the two sets $\{x \rightarrow x\}$ and $\{x \rightarrow y\}$ are incomparable. Using this algebra we can define the *valid shortest paths* algebra SA_{vsp} as the directed product of SA_{vp} and SA_{sp} ,

$$SA_{vsp} = \left(\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(\mathcal{F} \times \mathbb{R}_+), \mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty}), \emptyset, (\odot \times +), \text{id}_{\mathcal{P}(\hat{\mathcal{F}} \times \mathbb{R}_+^{\infty})} \right) \quad (54)$$

The set of signatures is the power set of all weighted composition functions and the set of labels is the power set of all weighted adaptation functions. Note that the order relation is defined on sets of weighted compositions. This order relation is the same as the partial order defined in Section 5.3 for the union-min operation.

Another idea is to define directly a Sobrinho's algebra with a specific weight function ω that maps a set of compositions to a set of weights. We define this algebra as follows,

$$SA_{vsp} = \left(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(\mathcal{F}), \mathcal{P}(\hat{\mathcal{F}}), \emptyset, \odot, \omega \right) \quad (55)$$

Where: $\omega : \mathcal{P}(\hat{\mathcal{F}}) \rightarrow \mathcal{P}(\mathbb{R}_+^{\infty})$ is the weight function.

5.6.2. Sobrinho's algebra over protocol stacks

Recall that the application of adaptation functions on a set of protocol stacks gives a new set of protocol stacks. This application can be used to define a Sobrinho's algebra over sets of protocol stacks with a specific operation to model the application of adaptation functions on protocol stacks. We define this algebra as follows,

$$SA_{vp} = \left(\mathcal{P}(\mathcal{H}), \subseteq, \mathcal{P}(\mathcal{F}), \mathcal{P}(\mathcal{H}), \emptyset, \otimes, \text{id}_{\mathcal{P}(\mathcal{H})} \right) \quad (56)$$

The operation $\otimes : \mathcal{P}(\mathcal{H}) \times \mathcal{P}(\mathcal{F}) \rightarrow \mathcal{P}(\mathcal{H})$ is the binary operation that models the application of a set of adaptation functions on a set of protocol stacks. The weight function ω is the identity function of the

Table 4
Algebras in the quadrants model.

weight computation	weight summarization algebraic	ordered
algebraic	Semiring	-
functional	Algebra of Functions	Sobrinho's Algebra

power set of all protocol stacks. Using this algebra, we define the *valid shortest paths* algebra by the following direct product,

$$SA_{vsp} = \left(\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(\mathcal{F} \times \mathbb{R}_+), \mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty}), \emptyset, (\otimes \times +), \text{id}_{\mathcal{P}(\mathcal{H} \times \mathbb{R}_+^{\infty})} \right) \quad (57)$$

The set of signatures is the power set of all weighted protocol stacks and the set of labels is the power set of all weighted adaptation functions. Note that the order relation is defined by a partial order on the sets of weighted protocol stacks.

Another possibility is to define directly a Sobrinho's algebra with a specific weight function that maps a set of protocol stacks to a set of weights. We define this algebra as follows,

$$SA_{vsp} = \left(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(\mathcal{F}), \mathcal{P}(\mathcal{H}), \emptyset, \otimes, \omega \right) \quad (58)$$

Where: $\omega : \mathcal{P}(\mathcal{H}) \rightarrow \mathcal{P}(\mathbb{R}_+^{\infty})$ is the weight function.

In Section 5.2, we have defined a binary concatenation operation of pairs of protocol stacks in order to model the transitive closure operation to compute valid paths by transitive closure. Using this operation, we can define a Sobrinho's algebra over the sets of all pairs of protocol stacks. This algebra is defined as follows,

$$SA_{vp} = \left(\mathcal{P}(\mathcal{H}^2), \subseteq, \mathcal{P}(\mathcal{H}^2), \mathcal{P}(\mathcal{H}^2), \emptyset, \diamond, \text{id}_{\mathcal{P}(\mathcal{H}^2)} \right) \quad (59)$$

Now we define the *valid shortest paths* Sobrinho's algebra on the pairs of protocol stacks by the following direct product,

$$SA_{vsp} = \left(\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \subseteq_{min}, \mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+), \mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty}), \emptyset, (\diamond \times +), \text{id}_{\mathcal{P}(\mathcal{H}^2 \times \mathbb{R}_+^{\infty})} \right) \quad (60)$$

The set of signatures and the set of labels are the power set of all weighted pairs of protocol stacks and the order relation is defined by a partial order on the sets of weighted pairs of protocol stacks.

Another idea is to define directly a Sobrinho's algebra with a specific weight function that maps a set of pairs of protocol stacks to a set of weights. We define this algebra as follows,

$$SA_{vsp} = \left(\mathcal{P}(\mathbb{R}_+^{\infty}), \subseteq, \mathcal{P}(\mathcal{H}^2), \mathcal{P}(\mathcal{H}^2), \emptyset, \diamond, \omega \right) \quad (61)$$

Where: $\omega : \mathcal{P}(\mathcal{H}^2) \rightarrow \mathcal{P}(\mathbb{R}_+^{\infty})$ is the weight function.

The most common violation among all of our Sobrinho's algebras is that the order relation is not total, as it is defined over subsets of the power set.

5.7. Properties of the algebras

The three algebras defined in this paper fit in the quadrants model of Gurney and Griffin (2007) as shown in Table 4. The first property to note is that the functional approach is more generic and powerful

than the algebraic approach. It provides the possibility to use various functions (which are not necessarily applied in the same way) instead of being limited to a single common operation for the path computation (i.e., such as \otimes). It is always possible to transform an algebraic approach into a functional one, but the reverse is not true. The second property concerns the selectivity. Algebras of functions and Sobrinho's algebras require selectivity, whereas semirings do not. In the case of Sobrinho's algebras, this implies totally ordered paths. For our three algebras, it is only possible to have a partial order relation (i.e., \subseteq_{min}).

These three algebras represent three different approaches which offer a variety of options available to the researcher, depending on the specific algebraic or functional usage as well as the nature of the operation itself. For example, the algebraic $+$ operation can be transformed in a functional one with the set of functions F_+ as shown in the previous definitions. However, applying a conversion on a stack is not the same operation as applying an encapsulation on this same stack, hence the use of the functional approach to represent the application operation of adaptation functions on stacks.

If one needs an algebra based on the application of adaptation functions on protocol stacks, then the functional approach is more appropriate (e.g., as is the case for the structures defined in Eqs. (48) and (57)). Otherwise, the algebraic approach will suffice (as is the case for the structures defined in Eqs. (30) and (42)). It can be noticed that the algebras of functions defined by Eqs. (44) and (51) is transformed from the semirings defined by Eqs. (30) and (42), and it uses function composition and stack concatenation. Therefore, there is a difference between an algebra of functions whose set of functions is obtained from a single algebraic operation, and an algebra of functions whose set of functions is a mix of adaptation functions which are applied on stacks in various ways.

Recall that a multilayer path is represented either by pair (composition of functions, weights) or by pair (concatenation of protocol stacks, weights). In this case, the algebra must contain a path extension operation composed of two operations. One for composing functions (or concatenating stacks) and one for adding weights. However, if the path weights are not calculated in the same way (i.e., based on several constraints and constants which can intervene depending on the use case), then a simple addition will not be enough. In this situation, Sobrinho's algebras are better suited because they offer the possibility of defining a specific weight function and not just a simple addition of weights along the path (e.g., as is the case for the structures defined in Eq. (55), 58 and 61).

Several examples leveraging Sobrinho's algebras include the cases where (i) we want to put constraints on the depth of tunnels with a specific weight, (ii) we want to favor flat paths rather than moving through tunnels with a weight factor, and (iii) we want to put specific weights for specific functions.

It should be reminded that in this paper, we studied the problem of valid shortest paths where the weight is defined as a simple and additive metric.

The transitive closure routing algorithm defined in Mouhoub et al. (2022a,b) derives from a transitive closure algebra which is algebraic and uses stack composition to compute the paths (as is the case for the structure defined in Eq. (42)) without having to rely on the use of adaptation functions. The stack vector algorithm defined in Lamali et al. (2019), however, derives from a stack vector algebra which is functional and uses the application of adaptation functions on the protocol stacks to compute the paths (as is the case for the structures defined in Eqs. (48) and (57)).

6. Convergence properties

In this section, we show some convergence properties on the routing algebras proposed in the previous section. Then, we prove the iterative convergence of the A^* matrix using a fixed point.

6.1. Monotonicity and isotonicity

In general, the convergence of routing protocols, e.g., distance-vector and path-vector protocols, is based on the two important properties of the corresponding routing algebras.

The first property, i.e., monotonicity, guarantees that the routing protocol converges in any network, but not necessarily to a global optimal solution, as shown in Sobrinho (2005).

Definition 8. Let \otimes be the extension operation and the order relation \leq_{\otimes} . The operation \otimes is *monotonic* if and only if,

$$a \leq_{\otimes} b \implies a \otimes c \leq_{\otimes} b \otimes c \quad \forall a, b, c \in S \quad (62)$$

We can see that all the proposed algebras in this paper do not satisfy this property. The composition of two sets can yield a new set (possibly the empty set) which is incomparable to the first sets. And the application of a set of adaptation functions on a set of protocol stacks can give a new set of protocol stacks (possibly the empty set) which is incomparable to the first set. For this, we prove the following proposition,

Proposition 4. The direct product operator $(\odot \times +)$ over the power set $\mathcal{P}(\hat{F} \times \mathbb{R}_+^{\infty})$ is not monotonic.

Proof. We prove the proposition by using the following counterexample of composition,

$$\left\{ (x \rightarrow xyx, 3), (y \rightarrow x, 2) \right\} (\odot \times +) \left\{ (x \rightarrow xy, 1) \right\} = \left\{ (x \rightarrow xx, 3) \right\} \quad (63)$$

We can see that the two sets $\{(x \rightarrow xyx, 3), (y \rightarrow x, 2)\}$ and $\{(x \rightarrow xx, 3)\}$ are not comparable by using the partial order relation of the union-min operation defined in Section 5.3. \square

We can use the same proof in the case of algebra of functions. For example,

$$F\left(\{(x, 1), (y, 2)\}\right) = \{(z, 3)\} \quad (64)$$

Where: $F = \{(x \rightarrow z, \omega \mapsto \omega + 2)\} \in \mathcal{P}(F \times F_+)$ is a set of a pair of adaptation function and a weight function. We can observe that this two sets of stacks are incomparable.

The second property, i.e., isotonicity, guarantees that the routing protocol converges to a global optimal solution, as proved by Sobrinho (2005).

Definition 9. Let \otimes be the extension operation and the order relation \leq_{\otimes} . The operation \otimes is *isotonic* if and only if,

$$a \leq_{\otimes} b \implies a \otimes c \leq_{\otimes} b \otimes c \quad \forall a, b, c \in S \quad (65)$$

In order to show that our algebras are isotonic, we prove the following proposition for semirings with composition. The same proof can be adapted to the algebras over protocol stacks.

Proposition 5. The direct product operator $(\odot \times +)$ over the power set $\mathcal{P}(\hat{F} \times \mathbb{R}_+^{\infty})$ is isotonic.

Proof. We prove the proposition by contradiction. Suppose there exists three sets of weighted composed functions \hat{F}_1 , \hat{F}_2 and \hat{F}_3 such that the following property is true,

$$\left(\hat{F}_1 \subseteq_{min} \hat{F}_2 \right) \wedge \left(\hat{F}_1 (\odot \times +) \hat{F}_3 \not\subseteq_{min} \hat{F}_2 (\odot \times +) \hat{F}_3 \right) \quad (66)$$

By the definition of the order relation of the union-min operation,

$$\hat{F}_1 \subseteq_{min} \hat{F}_2 \implies \exists \hat{F}, (\hat{F}_2 = \hat{F}_1^* \cup \hat{F}) \wedge (\hat{F}_1 \cup_{min} \hat{F}_1^* = \hat{F}_1) \quad (67)$$

Now, we compute the composition of \hat{F}_2 and \hat{F}_3 ,

$$\begin{aligned} \hat{F}_2(\odot \times +)\hat{F}_3 &= (\hat{F}_1^* \cup \hat{F})(\odot \times +)\hat{F}_3 = \\ &= (\hat{F}_1^*(\odot \times +)\hat{F}_3) \cup (\hat{F}(\odot \times +)\hat{F}_3) \end{aligned} \quad (68)$$

So, we can see that,

$$(\hat{F}_1(\odot \times +)\hat{F}_3) \subseteq_{\min} (\hat{F}_1^*(\odot \times +)\hat{F}_3) \cup (\hat{F}(\odot \times +)\hat{F}_3) \quad (69)$$

Hence, the sets \hat{F}_1 , \hat{F}_2 and \hat{F}_3 cannot exist. \square

6.2. Iterative convergence

It is known that in classical networks without absorbing circuits (i.e., with only positive weights) called free networks, the global optimal solution \mathbf{A}^* converges to the matrix $\mathbf{A}^{(n-1)}$ where $n-1$ is the maximum length of an elementary path, as proven by Carré (1971). This means that circuits only increase the weight of paths and therefore will be ignored by the computation. In our case, it is not the same situation, and as we saw in the multilayer model, circuits are allowed and sometimes necessary for some paths i.e., constructing the necessary protocol stack for the path to be valid. For this reason, we firstly define the multilayer circuit and the multilayer elementary path in the multilayer network model in order to generalize the convergence theorem cited above.

Definition 10. A multilayer path $p = h_i v_i f_i \dots v_{j-1} f_{j-1} h_j v_j$ is a *multilayer circuit* if and only if:

- The node v_i is the same node v_j , i.e., $v_i = v_j$
- The two stacks h_i and h_j received by v_i (or v_j) are the same, i.e., $h_i = h_j$

Using the above definition, we now define the multilayer elementary paths.

Definition 11. A *multilayer elementary path* is a multilayer path in which its circuits (if they exist) are non-multilayer circuits (i.e., such circuits start and end with a different protocol stack).

Definition 12. A *free multilayer network* is a network in which all of its circuits are non-multilayer circuits.

Lamali et al. (2018) proved some bounds on the *valid shortest paths* problem in a multilayer network modeled by a Push-Down Automaton (PDA). In this paper, we focus on the following proposition whose proof is given in Lamali et al. (2018). It is a proof similar to the one showing bounds on the length of the shortest word generated by a Context-Free Grammar (CFG) in Chomsky normal form as shown in Hopcroft and Ullman (1979):

Proposition 6. For any multilayer network \mathcal{N} , the valid shortest path (if any) between two nodes contains at most $O(2^t)$ links, where $t = (\lambda + 1)\lambda^2 n^2$.

Using this proposition, we prove the following theorem,

Theorem 1. In a free multilayer network \mathcal{N} we have:

$$\mathbf{A}^* = \mathbf{A}^{(k)} = \mathbf{I} \oplus \mathbf{A} \oplus \mathbf{A}^2 \oplus \dots \oplus \mathbf{A}^k \quad (70)$$

Where k is the maximum length of the multilayer elementary paths in \mathcal{N} , and it is equal to $2^{(\lambda+1)\lambda^2 n^2} - 1$.

Proof. Let \mathcal{N} be a free multilayer network with elementary paths of length at most k , $k \leq 2^{(\lambda+1)\lambda^2 n^2} - 1$. Let us suppose there is a integer $\ell > 0$, where $k + \ell$ is a fixed point for \mathbf{A}^* ,

$$\mathbf{A}^* = \mathbf{A}^{(k+\ell)} = \mathbf{I} \oplus \mathbf{A} \oplus \dots \oplus \mathbf{A}^k \oplus \mathbf{A}^{k+1} \oplus \dots \oplus \mathbf{A}^{k+\ell} \quad (71)$$

Let $\mathbf{A}_{i,j}^{(k+\ell)}$ be the set of valid weighted composed functions from node i to node j of size at most $k + \ell$, i.e., with at most $k + \ell$ adaptation

functions. Let us suppose there exists a valid path represented by a weighted composed function $(\hat{f}, \omega) \in \mathbf{A}_{i,j}^p$ of p adaptation functions such that $k + 1 \leq p \leq k + \ell$.

We say that (\hat{f}, ω) is optimal iff the following hypothesis is true:

Hypothesis: Let us assume there is no weighted composed function $(\hat{f}', \omega') \in \mathbf{A}_{i,j}^q$ with $1 \leq q \leq k$ such that $\hat{f}' = \hat{f}$ and $\omega' > \omega$. However, according to Proposition 3, if there is a valid shortest path with at least $k+1$ adaptation functions then there is another valid shortest path with at most k adaptation functions. More precisely, there exists a weighted composed function $(\hat{f}', \omega') \in \mathbf{A}_{i,j}^q$, such that $\hat{f}' = \hat{f}$ and $\omega' \leq \omega$. And by the definition of the operation \oplus (union-min), the composed function \hat{f} is ignored.

$$\{(\hat{f}, \omega), \dots\} \cup_{\min} \{(\hat{f}', \omega'), \dots\} = \{(\hat{f}', \omega'), \dots\} \quad (72)$$

Therefore, the valid path represented by \hat{f} is not elementary and it contains a multilayer circuit. Therefore, the multilayer network \mathcal{N} is not free. The above hypotheses does not hold, thus (\hat{f}, ω) cannot be optimal. Therefore, the maximum length k of elementary multilayer paths ($k \leq 2^{(\lambda+1)\lambda^2 n^2} - 1$) in a free multilayer network \mathcal{N} is a fixed point for the global optimal solution \mathbf{A}^* . \square

This proof is made in the case of the semiring with composition, however, the same proof can be adapted to the algebra of functions in which we replace the matrix product \mathbf{X}^k by the k applications of a set of functions \mathbf{A} on an initial matrix \mathbf{X} of protocol stacks.

6.3. Optimal solution size

Recall that in a multilayer network, there can be several valid shortest paths between a source and a destination with different compositions of adaptation functions which can induce different protocol stacks. In this situation, the number of valid shortest paths between each pair of nodes (v_i, v_j) depends on the number of possible protocol stacks in the network. In the following, we define this value in order to calculate the size of the global optimal solution \mathbf{A}^* of the shortest valid paths problem.

Let λ be the number of protocols and h_{max} be the maximum height of the protocol stack reached in a multilayer network. The number of possible protocol stacks is $\lambda^{h_{max}}$. This means that, for a pair of nodes (v_i, v_j) , the number of valid shortest paths with each possible protocol stack can be given by,

$$|\mathbf{A}_{i,j}^*| = \lambda + \lambda^2 + \lambda^3 + \dots + \lambda^{h_{max}} = \left(\frac{1 - \lambda^{h_{max}+1}}{1 - \lambda} - 1 \right) \quad (73)$$

Therefore, in the worst case, the number of valid shortest paths for all pairs is given by,

$$|\mathbf{A}^*| = n^2 \left(\frac{1 - \lambda^{h_{max}+1}}{1 - \lambda} - 1 \right) \quad (74)$$

This size can induce an exponential number of operations (composition of functions or concatenation of protocol stacks) in a multilayer routing algorithm. This bound is narrow, i.e., it is possible to find a network where the shortest possible path reaches a stack height of λn^2 protocols as proven by Lamali et al. (2019). Thus, the maximum length of a valid shortest path can be exponential as mentioned in the previous section. Finally, these theoretical limitations are not due to the multilayer routing algorithm but are inherent to the nature of the problem. In the next section, we show the value limits of each bound in practice.

7. Experimentation

We have performed simulations in order to validate the theoretical bounds given in the previous section. We have used the Stack-Vector (SV) algorithm and the Transitive-Closure (TC) algorithm. We did not allow multilayer circuits in the computation of the optimal solution. Additional results concerning these two multilayer routing algorithms

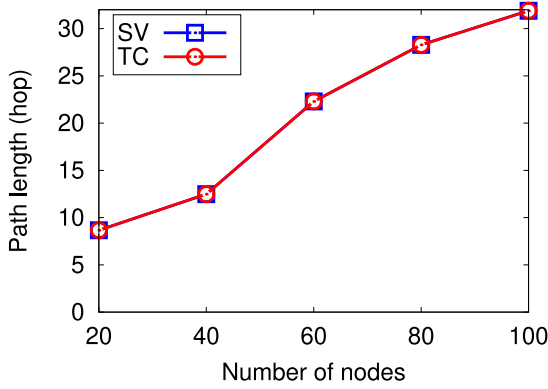


Fig. 5. Maximum path length for the optimal solution computed by the SV and TC algorithms vs network size.

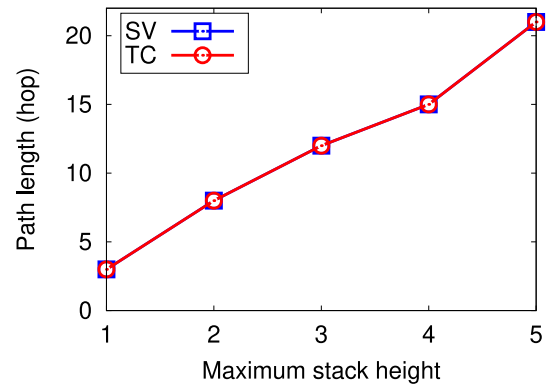


Fig. 6. Maximum path length for the optimal solution computed by the SV and TC algorithms vs maximum stack height (for a 100-node network).

are available for SV in Lamali et al. (2019) and TC in Mouhoub et al. (2022a).

The simulations were performed on a workstation running an hyperthreaded 8-core Intel i9-11900 processor able to reach 5.2 GHz with 128 GB of RAM. The implementation has been written in ISO C++14 with the help of the Igraph 0.8.0 library (Csárdi and Nepusz, 2005) for generating random multilayer network topologies. All the network topologies used for the experimentation are graphs randomly generated by a preferential attachment mechanism defined by Barabási and Albert (1999), where each newly added node is attached to 3 existing nodes. For a given number λ of protocols, there are $3\lambda^2$ possible adaptation functions. Each adaptation function is made available on a node with a fixed probability $p = 0.05$. The input parameters of each simulation are: the number of nodes n in the network and the number of protocols $\lambda = 2$. The output metrics are: the maximum path length, the maximum protocol stack height, the optimal solution size, and the algorithm efficiency. All the result values presented in the following figures are averaged over the result values of 100 runs.

7.1. Maximum path length

Fig. 5 shows the maximum path length obtained by the optimal solution computed by the SV and TC algorithms according to the network size. We can see that in all settings, the length of the computed paths never exceeds the maximum length of elementary multilayer paths shown above. This means that without multilayer circuits, the computation of the valid shortest path solution A^* converges after at most the maximum length of elementary multilayer paths as shown in Section 6. The results are the same for both the SV and the TC algorithms because they both compute the optimal solution.

As we have shown in the previous section, the maximum upper bound on the length of multilayer elementary paths depends on a formula including an exponent equal to $h_{max} + 1$ (this height is bounded in networks without multilayer circuits). Fig. 6 shows the maximum path length obtained by the optimal solution computed by the SV and TC algorithms according to the maximum stack height in a network of 100 nodes. We can see that the length of the computed paths never exceeds the maximum length of elementary multilayer paths shown in Section 6.

7.2. Maximum stack height

Fig. 7 shows the maximum stack height obtained by the optimal solution computed by the SV and TC algorithms according to the network size. The results are the same for both the SV and the TC algorithms because they both compute the optimal solution. These results show that the maximum stack height heavily impacts the maximum path length, and more precisely the convergence time and the size of the valid shortest path solution A^* .

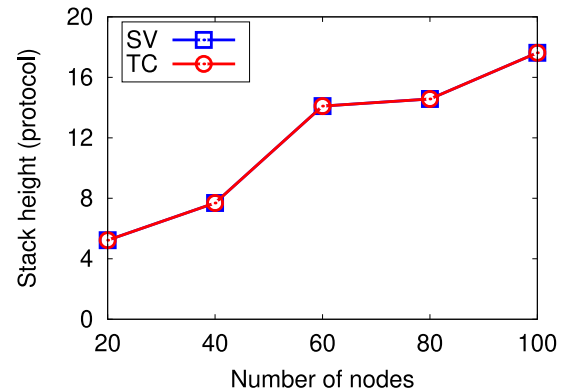


Fig. 7. Maximum stack height for the optimal solution computed by the SV and TC algorithms vs network size.

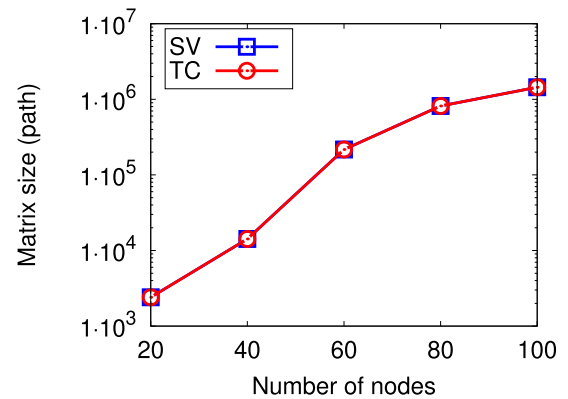


Fig. 8. Matrix size of the optimal solution computed by the SV and TC algorithms according to the network size.

7.3. Maximum routing tables size

Fig. 8 shows the size of the optimal solution (routing tables) computed by the SV and TC algorithms according to the network size. We can see that in all settings, the size of the computed solution never exceeds the maximum number of valid shortest paths, i.e., the A^* matrix size shown above. These results shown that the problem is complex and the optimal solution size can be large, especially if the maximum stack height is set to a high value. However, an optimal solution is still reachable in a multilayer network, by setting a small maximum stack height which ensures a limited amount of nested tunnels.

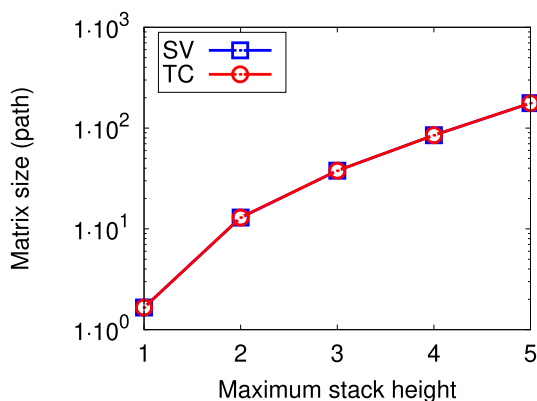


Fig. 9. Matrix size of the optimal solution computed by the SV and TC algorithms according to the maximum stack height in a network of 100 nodes.

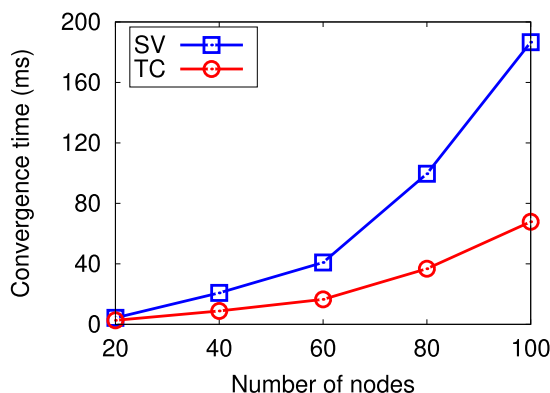


Fig. 10. The convergence time of the SV and TC algorithms according to the network size.

As we have shown in Section 6, the maximum size of the optimal solution A^* depends on an exponent equal to the maximum allowed height of the protocol stack (h_{max}) plus one. Fig. 9 shows the size of the optimal solution (which translates into routing table size) computed by the SV and TC algorithms according to the maximum stack height in a network of 100 nodes. We can see that the size of the computed solution never exceeds the maximum number of valid shortest paths, i.e., the A^* matrix size shown above.

7.4. Time efficiency of the algorithms

Fig. 10 shows the total time required by both the SV and the TC algorithms to compute the routing tables according to the network size. We call it the convergence time of the algorithm. We can see that the TC algorithm is more efficient than the SV algorithm, and can be up to 63% faster when executed on 100-node networks. These results show that even though these two algorithms are capable of finding the optimal solution for routing in multilayer networks (as proven by the algebras defined in the previous subsections), they can still exhibit disparate performances due to their different time complexities.

8. Real-world applications

Real-world cases where automatic routing in large scale multiprotocol networks could be leveraged include large IoT mesh and sensor networks where several protocols might be available and where network virtualisation is needed for multi-tenant scenarios. This can also be the case for 5G+ cellular networks which implement network slicing. Automatic tunneling could be very useful in mobile ad hoc networks

with high dynamics and in cellular networks with high end device mobility, as manually setting them up could be unmanageable.

Algorithms for multilayer network routing protocols such as Lamali et al. (2019), Mouhoub et al. (2022a), would typically occur when several virtual networks such as VPNs and/or protocols (such v4/v6) are layered one upon another. They could be used for intra-domain routing inside large telco networks. Automatic tunneling configuration depends on the available architecture. In a standard architecture, based on routers running fully distributed protocols, each router must run a specific daemon able to emit tunnel construction requests and able to receive and process such construction requests. In an SDN architecture, the controller is in charge of centrally computing dynamic tunnels as it knows the full network topology. Once defined, tunnel rules may be pushed in the related SDN switches for setting up the tunnels. Current algorithms are based on a proactive routing approach. This means that routing tables are filled and shall be updated (in case of network dynamics) before packets may travel from source to destination.

Multilayer inter-domain routing would require telecom operators to make multipartite agreements in order to share some parts of their multilayer routing information (links, tunnel endpoints) in order to improve the global routing efficiency of these algorithms. Operators not willing to share their adaptive function information could drastically reduce the number of valid paths available and could lead to a situation where no paths are available for some source-destination pairs.

9. Conclusion and future work

Routing algebras are powerful abstractions for studying the convergence properties of a routing protocol. They have been developed and used for a long time on classic routing protocols (e.g., Distance-Vector and Path-Vector). However, many networks today are heterogeneous and multilayered, allowing routing with automatic tunneling. In order to bridge this gap, we studied the application of routing algebras to heterogeneous and multilayer networks. For this, we defined three routing algebras for path computation with tunnels: a semiring, an algebra of functions and a so-called Sobrinho's algebra. We showed that the proposed routing algebras are isotonic and non-monotonic with a partial order. We also proposed a fixed point for these algebras and we proved the iterative convergence to the optimal solution of the *valid shortest paths (vsp)* problem.

For future work, we plan to study the asynchronous convergence of the stack-vector protocol by applying, if possible, the asynchronous convergence theorem of Daggitt et al. (2018) and implementing it with the Agda routing library (Daggitt et al., 2022). We also aim to use the lexicographic product defined by Gurney and Griffin (2007) and refined by Dynorowicz and Griffin (2013) to study various QoS metrics for Equal-Cost Multi-Path routing (ECMP), in the case of routing with automatic tunneling.

CRediT authorship contribution statement

Noureddine Mouhoub: Writing – original draft, Software, Formal analysis. **Maria Moloney:** Writing – review & editing, Methodology. **Damien Magoni:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization.

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.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank Géraud Sénizergues for his help on algebraic structures and axioms. This work was funded by The French National Research Agency - HÉRA project. Grant no.: ANR-18-CE25-0002.

Appendix A. Acronyms

Table 5 describes the meaning of various acronyms used throughout this paper.

Table 5
Acronyms for the Structures.

Acronym	Meaning
SR	Semiring structure
SR_{sp}	Shortest path semiring
SR_{vp}	Valid path semiring
SR_{vsp}	Valid shortest path semiring
AF	Algebra of functions structure
AF_{sp}	Shortest path algebra of functions
AF_{vp}	Valid path algebra of functions
AF_{vsp}	Valid shortest path algebra of functions
SA	Sobrinho's algebra structure
SA_{sp}	Shortest path Sobrinho's algebra
SA_{vp}	Valid path Sobrinho's algebra
SA_{vsp}	Valid shortest path Sobrinho's algebra

References

AL-Naday, M.F., Thomos, N., Reed, M.J., 2017. Information-centric multilayer networking: Improving performance through an ICN/WDM architecture. *IEEE/ACM Trans. Netw.* 25 (1), 83–97. <http://dx.doi.org/10.1109/TNET.2016.2571659>.

Amorim, K.S., Pavani, G.S., 2021. Ant colony optimization-based distributed multilayer routing and restoration in IP/MPLS over optical networks. *Comput. Netw.* 185, 107747. <http://dx.doi.org/10.1016/j.comnet.2020.107747>.

Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. *Science* 286 (5439), 509–512.

Botero, J.F., Molina, M., Hesselbach-Serra, X., Amazonas, J.R., 2013. A novel paths algebra-based strategy to flexibly solve the link mapping stage of VNE problems. *J. Netw. Comput. Appl.* 36 (6), 1735–1752. <http://dx.doi.org/10.1016/j.jnca.2013.02.029>.

Carré, B., 1971. An algebra for network routing problems. *J. Appl. Math.* 7 (3), 273–294. <http://dx.doi.org/10.1093/imamat/7.3.273>.

Carré, B., 1979. *Graphs and Networks*. Oxford University Press.

Csárdi, G., Nepusz, T., 2005. The igraph software package for complex network research. *J. Complex Syst.* 1695.

Daggitt, M.L., Griffin, T.G., 2018. Rate of convergence of increasing path-vector routing protocols. In: *IEEE Int'L Conf. on Network Protocols*, pp. 335–345.

Daggitt, M.L., Gurney, A.J.T., Griffin, T.G., 2018. Asynchronous convergence of policy-rich distributed Bellman-Ford routing protocols. In: *Int'L Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 103–116. <http://dx.doi.org/10.1145/3230543.3230561>.

Daggitt, M.L., Zimgrod, R., Griffin, T.G., 2022. Agda-routing for SIGCOMM 2018. <https://github.com/MatthewDaggitt/agda-routing/tree/sigcomm2018>. (Accessed: 2022-05-02).

Dynerowicz, S., Griffin, T.G., 2013. On the forwarding paths produced by internet routing algorithms. In: *2013 21st IEEE Int'L Conf. on Network Protocols*. ICNP, pp. 1–10. <http://dx.doi.org/10.1109/ICNP.2013.6733608>.

Etezadi, E., Beyranvand, H., Salehi, J.A., 2022. Latency-aware service provisioning in survivable multilayer IP-over-elastic optical networks to support multi-class of service transmission. *Comput. Commun.* 183, 161–170. <http://dx.doi.org/10.1016/j.comcom.2021.12.003>.

Gondran, M., Minoux, M., 1995. *Graphes Et Algorithmes*. Eyrolles, Paris, 3e édition.

Griffin, T.G., Sobrinho, J.L., 2005a. Metarouting. In: *Proceedings of the 2005 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '05, <http://dx.doi.org/10.1145/1080091.1080094>.

Griffin, T.G., Sobrinho, J.L., 2005b. Metarouting. *SIGCOMM Comput. Commun. Rev.* 35 (4), 1–12. <http://dx.doi.org/10.1145/1090191.1080094>.

Guo, C., Zhen, Y., cai Jia, G., wei Zhou, X., 2013. Multilayer satellite network routing based on hypergraph theory. In: *Int'L Conf. on Cyberspace Technology*. CCT 2013, pp. 312–315. <http://dx.doi.org/10.1049/cp.2013.2145>.

Gurney, A.J.T., Griffin, T.G., 2007. Lexicographic products in metarouting. In: *2007 IEEE Int'L Conf. on Network Protocols*, pp. 113–122. <http://dx.doi.org/10.1109/ICNP.2007.4375842>.

Halabi, W., Steenhaut, K., Nowé, A., Gurzi, P., 2011. Routing and signaling in GMPLS based DWDM multi domain multilayer networks using IP/WDM router. In: *2011 Eighth Int'L Conf. on Wireless and Optical Communications Networks*, pp. 1–5. <http://dx.doi.org/10.1109/WOCN.2011.5872921>.

Harden, J.C., 1949. *Direct and Semidirect Products of Semigroups* (Master's thesis). University of Tennessee - Knoxville, Knoxville.

Harhira, H.A., Pierre, S., 2008. A novel admission control mechanism in GMPLS-based IP over optical networks. *Comput. Netw.* 52 (6), 1281–1290. <http://dx.doi.org/10.1016/j.comnet.2008.01.004>.

Hebisch, U., Weinert, H., 1998. *Semirings: Algebraic theory and applications in computer science*. In: *Series in algebra*, World Scientific.

Hesselbach, X., Amazonas, J.R., Villanueva, S., Botero, J.F., 2016. Coordinated node and link mapping VNE using a new paths algebra strategy. *J. Netw. Comput. Appl.* 69, 14–26. <http://dx.doi.org/10.1016/j.jnca.2016.02.025>.

Hopcroft, J.E., Ullman, J.D., 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company.

Iqbal, F., van der Ham, J., Kuipers, F., 2015. Technology-aware multi-domain multi-layer routing. *Comput. Commun.* 62 (C), 85–96.

Katib, I., Medhi, D., 2011. A study on layer correlation effects through a multilayer network optimization problem. In: *Proceedings of the 23rd Int'L Teletraffic Congress*. ITC '11, Int'l Teletraffic Congress, pp. 31–38.

Khayou, H., Sarakbi, B., 2017. A validation model for non-lexical routing protocols. *J. Netw. Comput. Appl.* 98 (C), 58–64. <http://dx.doi.org/10.1016/j.jnca.2017.09.006>.

Kuipers, F., Dijkstra, F., 2009. Path selection in multi-layer networks. *Comput. Commun.* 32 (1), 78–85.

Lamali, M.L., Fergani, N., Cohen, J., 2018. Algorithmic and complexity aspects of path computation in multi-layer networks. *IEEE/ACM Trans. Netw.* 26 (6), 2787–2800. <http://dx.doi.org/10.1109/TNET.2018.2878103>.

Lamali, M.L., Fergani, N., Cohen, J., Pouyllau, H., 2016. Path computation in multi-layer networks: Complexity and algorithms. In: *IEEE Conf. on Computer Communications*.

Lamali, M.L., Lassourreille, S., Kunne, S., Cohen, J., 2019. A stack-vector routing protocol for automatic tunneling. In: *IEEE Conf. on Computer Communications*, pp. 1675–1683. <http://dx.doi.org/10.1109/INFOCOM.2019.8737536>.

Lamali, M.L., Pouyllau, H., Barth, D., 2013. Path computation in multi-layer multi-domain networks: A language theoretic approach. *Comput. Commun.* 36 (5), 589–599.

Minoux, M., 2001. *Graphes, Dioides et Semi-Anneaux: Nouveaux modèles et algorithmes*. TEC and DOC.

Minoux, M., Gondran, M., 2008. *Graphs, dioids and semirings. New models and algorithms*. Operations Research/Computer Science Interfaces Series, vol. 41, Springer, <http://dx.doi.org/10.1007/978-0-387-75450-5>.

Mohri, M., 2002. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.* 7 (3), 321–350.

Mouhoub, N., Lamali, M.L., Magoni, D., 2022a. A highly parallelizable algorithm for routing with automatic tunneling. In: *IFIP Networking Conf.*, pp. 1–9. <http://dx.doi.org/10.23919/IFIPNetworking55013.2022.9829784>.

Mouhoub, N., Lamali, M.L., Magoni, D., 2022b. (Presque) Floyd-Warshall sous stéroïdes : (encore) un algorithme de routage pour l'établissement automatique de tunnels. In: *24èmes Rencontres Francophones sur Les Aspects Algorithmiques Des Télécommunications*. Saint-Rémy-Lès-Chevreuse, France.

Mouhoub, N., Lamali, M.L., Magoni, D., 2022c. Semiring algebraic structure for metarouting with automatic tunneling. In: *1st IEEE Int'L Workshop on New IP and beyond (Co-Located with the 30th IEEE ICNP)*, pp. 1–6. <http://dx.doi.org/10.1109/ICNP55882.2022.9940413>.

Raste, T.M., Kulkarni, D., 2008. Design and implementation scheme for deploying IPv4 over IPv6 tunnel. *J. Netw. Comput. Appl.* 31 (1), 66–72. <http://dx.doi.org/10.1016/j.jnca.2006.06.009>.

Ricciardi, S., Palmieri, F., Castiglione, A., Careglio, D., 2015. Energy efficiency of elastic frequency grids in multilayer IP/MPLS-over-flexgrid networks. *J. Netw. Comput. Appl.* 56, 41–47. <http://dx.doi.org/10.1016/j.jnca.2015.06.014>.

Sobrinho, J.L., 2003. Network routing with path vector protocols: Theory and applications. In: *Proceedings of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '03, Association for Computing Machinery, New York, NY, USA, pp. 49–60. <http://dx.doi.org/10.1145/863955.863963>.

Sobrinho, J., 2005. An algebraic theory of dynamic network routing. *IEEE/ACM Trans. Netw.* 13 (5), 1160–1173. <http://dx.doi.org/10.1109/TNET.2005.857111>.

Sobrinho, J.L., Ferreira, M.A., 2020. Routing on multiple optimality criteria. In: *Proceedings of the Annual Conf. of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM '20, Association for Computing Machinery, New York, NY, USA, pp. 211–225. <http://dx.doi.org/10.1145/3387514.3405864>.

Tseng, C.C., Lin, C.L., Yen, L.H., Liu, J.Y., Ho, C.Y., 2013. Can: A context-aware NAT traversal scheme. *J. Netw. Comput. Appl.* 36 (4), 1164–1173. <http://dx.doi.org/10.1016/j.jnca.2013.02.001>.

- Urta, A., Calle, E., Marzo, J.L., Vila, P., 2007. An enhanced dynamic multilayer routing for networks with protection requirements. *J. Commun. Netw.* 9 (4), 377–382. <http://dx.doi.org/10.1109/JCN.2007.6182872>.
- Van Mieghem, P., Kuipers, F., 2004. Concepts of exact QoS routing algorithms. *IEEE/ACM Trans. Netw.* 12 (5), 851–864.
- Wu, X., Wang, J., Wang, P., Bian, Z., Huang, T., Guo, Y., Fujita, H., 2021. Trustworthiness assessment for industrial IoT as multilayer networks with von Neumann entropy. *Appl. Soft Comput.* 106, 107342. <http://dx.doi.org/10.1016/j.asoc.2021.107342>.
- Xingfu, Z., Deqiang, D., 2021. Adaptive routing planning in multi layer and multi domain optical networks. In: *Int'l Conf. on Communications, Information System and Computer Engineering*. pp. 821–825. <http://dx.doi.org/10.1109/CISCE52179.2021.9445973>.

Noureddine Mouhoub: is a computer science Ph.D. student and teaching assistant at the University of Bordeaux and at LaBRI since 2019. Before that he obtained a master's degree in theoretical computer science at the University of Bordeaux in 2019. His main research interests are: heterogeneous and multilayer networks, automatic tunneling, path computation, routing algebras, and distributed algorithms, in particular routing with automatic tunneling in heterogeneous and multilayer networks.

Dr Maria Moloney: works in the School of Computer Science in University College Dublin and is currently collaborating with the Centre for Cybersecurity and Cybercrime Investigations (CCI) there. Prior to this, she was Head of Research at Escher Group

Holdings Plc, the leading software supplier to the postal industry worldwide. She was awarded a Government of Ireland Research Scholarship under the Embark Initiative to pursue her Ph.D. in Information Systems at Trinity College Dublin, which she completed in 2014. Her research interests lie in the areas of Data Protection, Privacy by Design, Security by Design and Cybersecurity. Her work has been published in journals such as the *International Journal of Industrial Ergonomics* and presented at prestigious conferences such as the *International Conference on Information Systems*. She was a Fulbright TechImpact Scholar, and studied cybercrime and criminal behavior in online settings in Maryland University in 2018/2019.

Prof. Damien Magoni: received the M.Eng. degree from Télécom Paris in 1995, and the M.Sc. and Ph.D. degrees from the University of Strasbourg, in 1999 and 2002, respectively. He has been a Visiting Researcher at various institutions around the world, including the AIST at Tsukuba, the University of Sydney, the University of Michigan at Ann Arbor, University College Dublin and the University of Luxembourg. He has been a Full Professor of computer science at the University of Bordeaux, since 2008. From 2002 to 2008, he was an Associate Professor at the University of Strasbourg. Part of his research has been supported by grants from the European Union, the CNRS, and Science Foundation Ireland. He has co-published 90+ refereed research papers. He also has authored several open-source software for networking research and teaching. His latest contributions are the virtual network device and the network mobilizer, which jointly enable the emulation of mobile networks. His main research interests include computer communications and networking, with a focus on internet architecture, protocols, and applications. He has reviewed for 20+ academic journals and has been in the TPC of 30+ high-level conferences. He is a Senior Member of the ACM and the IEEE.