A Theoretical Framework for Network Monitoring Exploiting Segment Routing Counters

Marco Polverini, Antonio Cianfrani, *Member, IEEE*, and Marco Listanti, *Member, IEEE*

***Abstract*—Self-driving networks represent the next step of network management techniques in the close future. A fundamental point for such an evolution is the use of Machine Learning based solutions to extract information from data coming from network devices during their activity. In this work we focus on a new type of data, available thanks to the definition of the novel SRv6 paradigm, referred to as SRv6 Traffic Counters (*SRTCs*). *SRTCs* provide aggregated measurements related to forwarding operations performed by SRv6 routers. In this work a detailed description of different *SRTCs* types (SR.INT, PISD, PSID.TM and POL) is provided and their relationships is formalized. The theoretical framework deployed is used to identify, on the basis of network configuration parameters of both SRv6 and IGP protocols, the minimum set of independent *SRTCs* to characterize the *Network Status*: we show that about the 80% of counters can be neglected with no information loss. We also apply our framework to two use cases: i) Traffic Matrix (TM) Assessment and ii) Traffic Anomaly Detection. For the TM assessment, we show that in a partially deployed SRv6 scenario a specific type of *SRTCs*, i.e., PSID, is more reliable than other ones; on the contrary, in a fully deployed scenario POL and PSID.TM counters provide the full TM knowledge. For the Traffic Anomaly Detection case, we show that known solutions based on link load measurements can be improved when integrating *SRTCs* information.**

***Index Terms*—Segment routing, traffic matrix computation, interface counters, traffic anomaly detection.**

I. Introduction

IN THE era Artificial Intelligence, Networking Monitoring is facing a significant change with respect to standard solutions used by network operators. The possibility of obtaining real time operational data from network devices allows the Network Monitoring to evaluate the Network Status and trigger specific operational procedures, such as re-routing in case of traffic congestion or maintenance in case of devices outage. In the last years, the network operators are looking for the implementation of self-driving networks based on advanced Machine Learning solutions, so that to provide autonomous operations such as configuration and management [1].

A key aspect for new management solutions is the availability of different data from the network: elaborating a significant amount of data coming from different sources is the only way to really exploit the capabilities of Machine Learning based solutions. Looking at operational network data provided by devices (mainly routers), two different kind of data can be collected: i) flow level measurements data and ii) aggregated measurements data. Flow level data requires the use of a dedicated monitoring protocol (such as IPFIX and Cisco Netflow) able to perform packets gathering at network devices and to collect the distribute measurements in a central node. Aggregated measurements are available in all routers as an extra-information provided by usual operational procedures.

We focus on aggregated measurements since they are available in all network devices with no need of dedicated protocols. The most known type of aggregated measurement is the link load, that all IP routers compute when performing forwarding operations on received traffic. In a Software Defined Network (SDN) scenario, the SDN switches also provide advanced counters: matching rules counters, able to measure traffic matching a specific rule of the SDN forwarding table [2]. In this work we focus on novel aggregated measurements available in the Segment Routing version 6 (SRv6) network paradigm. SRv6 is a source routing solution based on the use of Segment Lists: each network path is represented by an ordered list of Segments, where each Segment is the identifier of a node to be crossed. The paths among Segments are computed by classical IP routing protocols. SRv6 is supported today by about 25 different hardware platforms from companies such as Cisco, Huawei, Broadcom and others. It is also supported by two open source projects; the SRv6 Linux Kernel implementation and the FD.IO VPP one. The aggregated measurements of SRv6, referred to as SRv6 traffic counters (*SRTCs*), provide a finer granularity with respect to traffic load. Four different types of *SRTCs* have been defined [3]: i) *per-interface* counters (SR.INT), ii) *prefix-SID* counters (PSID), iii) *Traffic Matrix* counters (PSID.TM), and iv) *SR Policy* counters (POL).

The main aim of this work is to provide a theoretical framework for the characterization of *SRTCs* contribution to the Network Status evaluation. The framework provides guidelines to detect relationships among different *SRTCs* on the basis of network paths and Segment Lists structure. In this way we provide a simple mechanism for the detection of the minimum set of useful *SRTCs* able to fully characterize the Network Status; this allows a significant amount of counters to be neglected, speeding up the data collecting phase performed by the Monitoring tool. As examples of the application of the proposed framework, we investigate two different use

Manuscript received December 13, 2019; revised May 7, 2020; accepted June 3, 2020. Date of publication June 11, 2020; date of current version September 9, 2020. The associate editor coordinating the review of this article and approving it for publication was K. Shiomoto. *(Corresponding author: Marco Polverini.)*

The authors are with the Department of Information engineering, Electronics and Telecommunications, University of Roma "Sapienza," 00184 Roma, Italy (e-mail: marco.polverini@uniroma1.it; antonio.cianfrani@uniroma1.it; marco.listanti@uniroma1.it).

Digital Object Identifier 10.1109/TNSM.2020.3001809

cases that can be greatly impacted by *SRTCs*: i) Traffic Matrix Assessment (TMA) and ii) Traffic Anomaly Detection. We focus on the advantages provided by *SRTCs* when determining the TM of a network, especially in a partial deployment scenario; moreover, we show that *SRTCs* can greatly help standard anomaly detection solutions thanks to their thinner granularity with respect to monitoring data.

The main contributions of the paper are listed below:

- this is the first work, after its preliminary conference version [4], providing a theoretical evaluation of the *SRTCs* contribution to the Network Status characterization;
- it provides general guidelines, based on the network configuration parameters, for the identification of relationships among different SR counters; in this way it is possible to speed up the collection phase or to detect anomalous network behaviors;
- it provides insights for the use of *SRTCs* in the Traffic Matrix Assessment use case; it is shown that in a full SRv6 scenario the TM can be easily measured by POL counters, while in the case of a partial deployment (i.e., when not all *SRTCs* are available in all network nodes) the PSID counters provide more information than PSID.TM and POL counters;
- it proves that *SRTCs* are helpful when performing Traffic Anomaly Detection use case; the performance of classical solutions, such as Fourier analysis (FFT) and Principal Component Analysis (PCA), can be improved making also possible the quantification of the anomaly size.

The paper is organized as follows: an introduction on the SRv6 architecture is proposed in Section II, while the SRv6 traffic counters are introduced in Section III. System model and problem statement are reported in Section IV-A. In Section V we characterize the relations arising between the information achieved by the different types of SRv6 traffic counters. In Section VI is reported a deep performance evaluation, to show the potential benefit achieved by the use of SRv6 traffic counters for the TM assessment and traffic anomaly detection, the review of the state of the art in traffic measurement and monitoring is reported in Section VII. Finally, future work and conclusion are given in Section VIII.

II. Segment Routing Background

Segment Routing (SR) [5] is a novel network paradigm based on explicit source routing: the SR source node of a packet select the network path to be followed and encodes it into the packet header. In SR, end-to-end paths are encoded as an ordered list of instructions, also referred to as *Segments*; the full list of segments is named *Segment List* (*SID list*). Each segment may have either a topological meaning (e.g., send the packet to a given node) or service instruction (e.g., duplicate the packet). Segments are expressed as labels, named *Segment IDentifiers* (SIDs). SR allows to create an overlay network by means of SR tunnels, encoded as *SID lists*, on top of an underlay network; two different underlay solutions, also known as data plane, have been considered, i.e., MPLS and IPv6. Depending on the used data plane, a SID has a different physical structure. Specifically, in SR over MPLS (SR-MPLS) [6], SIDs are encoded as MPLS labels, while in SR over IPv6 (SRv6) [7], SIDs are expressed as IPv6 addresses. The present work is focused on SRv6 networks.

Each SID [8] is composed by three different parts: i) a *locator*, ii) a *function*, and iii) a set of *args bits*. The *locator* refers to the node where the SID is instantiated, the *function* specifies what action the node must perform, and the *args bits* allow to pass an input to the function. While the *locator* and the *function* are mandatory, the *args bits* are optional. The *locator* is represented by an IPv6 prefix, and is referred to as *prefix-SID*. In general, the *locator* part of the SID is routable, i.e., every router has a forwarding rule to reach the targeted prefix. Generally, in an SR domain, a locator is assigned to each SR capable router, i.e., a router supporting SRv6 forwarding functions. SRv6 packet forwarding works as follows: when a border router receives a packet, a specific *SID list* chosen according to a given *SR Policy* is assigned to it. An *SR Policy* is composed by a *match* and an *action*: the first specifies the traffic flows to be processed through the specific *SR Policy*, while the second specifies the *SID list* to be used. The node where the *SR Policy* is installed is referred to as head end node [5]. Specifically, each node of the SR domain maintains an *SR Policy* table that stores all the available policies at that node. The outer header has an SRH containing the *SID list* and a *segment left* (sl) field. The sl is a pointer to the active segment (*actsgm*), i.e., the current instruction to be applied to the packet. The *actsgm* is also inserted in the destination address field of the outer IPv6 header. Transit routers forward incoming packets by inspecting the IPv6 destination address of the outer header. When a node receives a packet having as *actsgm* its locator, the so called *MyLocalSID* table is inspected, in order to detect the function to be executed. After executing the function, the *actsgm* has to be updated: the *sl* is decremented and the next segment of the *SID list* is copied into the IPv6 destination header assuming the role of new *actsgm*. Finally, before the packet leaves the SR domain, the SRv6 encapsulation must be removed.

III. Segment Routing Traffic Accounting Counters

SR Traffic Counters (*SRTCs*), described in [3], are implemented in SR capable nodes to account incoming SR packets (or bytes). In this section, an overview of the mechanisms used to update the *SRTCs* is provided. In order to make the explanation easier to follow, Fig. 1 shows an example of a simple reference scenario.

In Fig. 1 a SRv6 domain with 9 SR capable nodes is reported. Nodes 1, 4, 6 and 9 represents the border of the SR domain (represented by the black dashed line in the figure): in other words, packets coming from different domains must be classified and encapsulated using dedicated SR Policies (policies are not shown in Fig. 1). In the configuration of the SR domain, the Network Operator can delimit the region of interest[1] of the network, configuring the so-called *TM Border*.

[1] For whatever reason (troubleshooting, traffic analysis, etc.), the Network Operator might be interested in monitoring a given portion of the infrastructure.

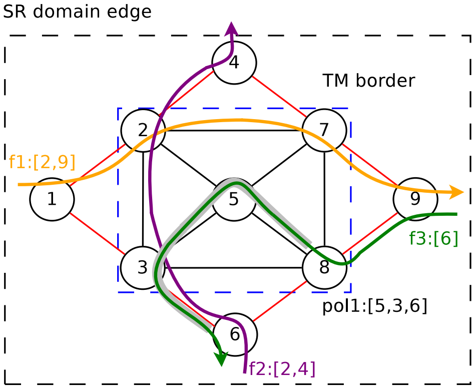


Fig. 1. Example of how *SRTCs* are updated.

The *TM Border* is used only for the definition of specific SR counters (as explained in the following) and it has no impact on network functioning. The *TM Border* can be equivalent to the overall SR domain or can be part of it. As an example, in Fig. 1 the *TM Border* is part of the SR domain, and it is delimited by a blue dashed line. According to the definition of the *TM Border*, the interfaces of the routers can be classified into two main groups: *internal* and *external*. An external interface connects a node external to the *TM Border* to a TM border node, whereas an internal interface connects two nodes of the *TM Border* or internal to it. In Fig. 1, the external interfaces are the ones connected to links shown in red, while the internal interfaces are those connected to links depicted in black. A link connected to an external interface is referred to as external link. Moreover, three traffic flows and the related *SID lists* are highlighted using different colours (no other traffic flows exist in the network). The instruction related to each SID implies the use of the shortest path toward the specified node (assuming that all links have equal IGP weight). Finally, an *SR Policy* is defined at node 8, to be applied to packets of flow $f3$.

Four different types of *SRTCs* are defined in [3]: i) *per-interface* counters (SR.INT), ii) *prefix-SID* counters (PSID), iii) *Traffic Matrix* counters (PSID.TM), and iv) *SR Policy* counters (POL).

Per-Interface SR Counters (SR.INT) account the packets belonging to the aggregated SR traffic sent or received through a specific interface. Each node maintains two SR.INT counters (TX/RX) for each interface. Referring with (i, j) to the interface connecting the node i to node j, the related counter is indicated as SR.INT(i, j). In the example of Fig. 1, SR.INT(2, 7) accounts the packets of the flow $f1$.

Per Prefix-SID SR Counters (PSID) account the packets belonging to SR traffic according to the *actsgm*.[2] In particular, every node maintains a PSID for each prefix SID existing in the considered SR domain. Considering node n and prefix-SID S, the related counter is referred to as PSID(n, S), and accounts the packets of the aggregated amount of traffic received by node n and having the prefix-SID S as *actsgm*. With reference to the example of Fig. 1 and focusing on PSID(2, 2), it is accounting the total number of packets of the flows $f1$ and $f2$.

Traffic Matrix SR Counters (PSID.TM) account the packets of the aggregated SR traffic characterized by a specified couple of *actsgm* and the type of interface on which the traffic is received. Specifically, PSID.TMs act as the PSIDs, but are updated only if the traffic is received from an interface marked as external. Then, considering node n and prefix-SID S, the related counter is referred to as PSID.TM(n, S), and accounts the aggregated amount of traffic received by node n on a external interface and having the prefix-SID S as *actsgm*. As an example, PSID.TM(2, 2) of the network scenario reported in Fig. 1 only accounts flow $f1$. In fact, despite both $f1$ and $f2$ have 2 as *actsgm* when reacheved by node 2, the latter is received from an internal interface. As a consequence, $f2$ is accounted in PSID(2, 2) but not in PSID.TM(2, 2).

Per Policy SR Counters (POL) account the aggregated SR traffic that is steered through a pre-configured SR tunnel. This type of counter is maintained only at the head end node of the considered *SR Policy*. Considering a generic policy π, POL(π) accounts the aggregated traffic steered through π. Considering Fig. 1, POL(pol1) defined at node 8 accounts the traffic flow $f3$, which is steered by using the policy pol1.

With respect to traffic counters available in today networks, *SRTCs* provide a different traffic aggregation level. In a classical IP network, the routers perform traffic accounting on their interfaces: more in detail, each interface counts the amount of packets/byte received and/or forwarded. The traffic is not differentiated on the basis of the IP destination addresses and so the information provided has an high aggregation level. The IP per-interface counters are available also in SR routers and are referred to as SR.INT in our evaluation.

In the case of MPLS networks, the traffic counters available allow for a fine grained accounting of network traffic. The availability of a dedicated MPLS tunnels for each TE path makes possible to account the traffic of each LSP [9]. Anyway, the use of tunnels is also a huge limitation for MPLS in terms of scalability. Segment Routing is able to overcome MPLS limitation, realizing TE paths with no need of dedicated tunnels. In this way, SR is not providing traffic counters for each traffic relationship and thus the definition of a novel traffic accounting strategy is needed and motivate our work.

The actual implementations of SRv6 can have a full support of *SRTCs*, as for last Cisco devices [10], or a partial one, such as for software implementation. We investigated the FD.IO VPP platform, and we checked that PSID counters and an aggregation of all policies counters are only available. The partial support is mainly due to the complexity of the *SRTCs* accounting procedure at data plane, that can be a limiting factor for performance of software implementations and old hardware devices.

IV. Modeling the Network Status Through SRv6 Counters

In this section a theoretical framework to model the behavior of *SRTCs* is defined. The main goal of the proposed framework is to detect the relationships among *SRTCs* values, the current

[2] If the incoming packet is not SRH encapsulated, the locator of the egress node is considered [3].

routing configuration and the volume of traffic flows. The set of all the previous elements is referred to as *Network Status*. In Section VI, meaningful case studies (traffic matrix assessment and traffic anomaly detection), are faced by analyzing the *Network Status*. To simplify the description, the *Network Status* is evaluated starting from SR.INT counters, and then its extension considering the different counters (PSID, PSID.TM and POL) is provided.

A. System Model

The reference scenario is the one of an SR overlay built on top of an IPv6 underlay. In the underlay, an IGP protocol is used: i) to distribute the SIDs[3] availability information among nodes, and ii) to compute underlay shortest paths. The underlay network is represented by the direct graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of *N* nodes and $\mathcal{L}$ is the set of *L* links, respectively. Traffic flows cross the SR domain from an Ingress nodes to an Egress nodes. Globally, there are *K* flows traversing the considered SR domain. Traffic flows are represented by a vector $\mathbf{x}$, and its generic element x_k $(k = 1 \ldots K)$ is the intensity of the *k*-th flow.

The set of *SR Policy* defined within the considered SR domain is named Π. Each policy $\pi_j \in \Pi$ is fully described by the following three parameters: i) the locator of the head end node *I*, ii) the *color*, representing the intent (e.g., low-latency) of the considered policy, and iii) the locator of the end point node *E*. In short, an *SR Policy* is identified by the 3-tuple $\pi_j = (I, color, E)$. The *SID list* associated to the policy π_j is referred to as $\mathcal{S}_j$. SR policies can be defined in every node of the SR domain. All incoming flows must be "intercepted" by a dedicated policy in the Ingress nodes, before being forwarded into the SR domain. The function $\phi(k) : 1 \ldots K \rightarrow \Pi$, returns the policy π_j used to steer the flow *k* into the SR domain.

The routing of each flow depends on two different types of paths: i) the overlay paths and ii) the underlay paths. Following the SRv6 architecture specifications, each flow is steered according to a *SID list*, which specify the ordered list of locators to cross, while the path from a locator to the next one is forced by the underlay IGP routing. The underlay paths are fully described by *G* matrix, having *L* rows and $N(N-1)$ columns (one for each pair of nodes in the network). Considering the *c*-th column of the matrix *G*, it represents the underlay path found by the IGP protocol between the *c*-th pair of nodes. The function $\eta(i, j) : \mathcal{N} \times \mathcal{N} \rightarrow 1 \ldots N(N-1)$ is defined to keep the mapping between the columns of the matrix *G* and the related pair of nodes. Specifically, it takes as input a pair of nodes (their locators) and returns as output the ID of the column of the matrix *G* describing the underlay path between these two nodes. The element $g_{l,c}$ represents the fraction of traffic crossing link *l* if the *c*-th underlay path is considered to steer one unit of traffic. Since ECMP is supported by IPv6, the elements $g_{l,c}$ are real numbers ranging between 0 and 1. The overlay path are described by *R* matrix, having *L* rows and *K* columns. The element $r_{l,k}$ is equal to the percentage of the *k*-th flow that is routed over link *l*. Considering that SRv6 exploits IGP paths, also the elements $r_{l,k}$ are real numbers ranging between 0 and 1. The notation $\mathbf{r}_k$ (column vector) is used to refer to the overlay path followed by the *k*-th flow.

Clearly, there is a strong relationship among the *R* and *G* matrices. In particular, each element of the matrix *R* can be represented as a linear combination of the elements of *G*. Considering flow *k* and its *SID list* $\mathcal{S}_{\phi(k)} = \{I, i_1, \ldots, i_{S-1}, E\}$,[4] where *I* and *E* are the SIDs of the head end and end point nodes of the policy used to steer the flow *k* and *S* is the total length of the *SID list*, for each link *l* the following condition holds:

$$r_{l,k} = \sum_{i=1}^{S-1} g_{l,\eta(\mathcal{S}_{\phi(k)}[i],\mathcal{S}_{\phi(k)}[i+1])} \ \forall l \in \mathcal{L}. \tag{1}$$

In Eq. (1) the notation $\mathcal{S}_{\phi(k)}[i]$ is used to identify the *i*-th locator of the *SID list*. Eq. (1) shows that the overlay path followed by the *k*-th flow, i.e., the *k*-th column of the matrix *R*, can be represented as the concatenation of underlay paths. As an example, in case of the flow $f1$ of Fig. 1, the overlay path of the orange flow is represented by the sequence of links (1, 2)−(2, 7)−(7, 9), that is the concatenation of the two underlay paths $\mathbf{g}_{\eta(1,2)}$, composed by link (1, 2), and $\mathbf{g}_{\eta(2,9)}$, composed by links (2, 7) and (7, 9).

Using the previous notation, it is possible to formally compute the amount of traffic crossing each link *l*, referred to as $y_L(l)$. This quantity corresponds to the measurement performed by the SR.INT counter related to link *l*. The $y_L(l)$ values can be collected into the vector $\mathbf{y_L}$, whose *l*-th element represents the amount of traffic flowing over the link *l*. Considering the vector of the measurements performed by the SR.INT counters, the following equation holds:

$$\mathbf{y_L} = R \cdot \mathbf{x} \tag{2}$$

Specifically, Eq. (2) shows that the amount of traffic carried by a link *l* is given by the weighted summation of the flows routed over the considered link. Eq. 2 represents the *Network Status* according to SR.INT counters, i.e., the mathematical relationship among counters, intensity of traffic flows and routing. In the next, the model is extended by including the remaining *SRTCs*.

B. Integrating SRTCs in the Network Status

The linear system reported in Eq. (2) has a number of equations equal to the number of links. Specifically, every measurement performed by an INT.SR counter allows for the definition of an equation of the *Network Status*. Similarly, the other types of *SRTCs* provide additional equations (one for each counter) to extend the system reported in Eq. (2).

As first, the contribution of the PSID counters is investigated. The number of different PSID counters to be considered for each node is equal to the number of different locators in

[3] In the following, the terms locator and SID assume the same meaning and are used interchangeably.

[4] For the ease of presentation it is assumed that the first SID in a *SID list* is always the one representing the head end node, and the first *actsgm* is the second SID in the *SID list*.

the SR domain. As a consequence, the number of PSID counters in the SR domain is equal to N^2.[5] Looking at the *Network Status*, N^2 new equations are added to the linear system shown in Eq. (2). The value of the PSID(*i, a*) counter is denoted to as $y_B(i, a)$. For each PSID counter, the row vector $\mathbf{b^{(i,a)}}$ is defined, having length equal to *K*: its generic element $b_k^{(i,a)}$ is the percentage of flow *k* that crosses node *i* having a as active segment. The coefficients $b_k^{(i,a)}$ can be computed from the matrix *G* and the set of segment lists. The full collection of all the $\mathbf{b^{(i,a)}}$ vectors leads to the definition of the matrix *B* (of dimension $N^2 \times K$), that represents the system of linear equations that can be written using the information achieved by the introduction of the PSID counters. Similarly, the vector $\mathbf{y_B}$ collects all the measurements got by means of the PSID counters. The relation that holds between the vector **x** and the vector that collects the measurements of the PSID counters is expressed as:

$$\mathbf{y_B} = B \cdot \mathbf{x} \tag{3}$$

Similar consideration can be repeated for the PSID.TM counters. The notation $y_M(i, a)$ is used to refer to the value of the PSID.TM counter at node *i* for the active segment a. The overall number of PSID.TM counters in the network is equal to N^2.[6] A row vector $\mathbf{m^{(i,a)}}$ is used to report the percentage of each flow reaching node *i* on an external interface and having a as *actsgm*. Defining *M* the matrix that collects all the vectors $\mathbf{m^{(i,a)}}$ (dimension $N^2 \times K$), the relationship among routing paths and PSID.TM counters can be expressed as follows:

$$\mathbf{y_M} = M \cdot \mathbf{x} \tag{4}$$

The last contribution to be exploited is the one achieved by the POL counters. The set of equations added to the *Network Status* model by the availability of the POL counters is represented by means of the matrix *P*. *P* has a number of rows equal to the size of the set Π, and a number of columns equal to the number of flows (*K*). Each element $p_{j,k}$ of the matrix *P* is a binary value stating if the flow *k* hits or not the *SR Policy* π_j. Furthermore, the value of the POL(π_j) is denoted as $y_P(\pi_j)$. Then, the relationship among the value of the POL counters and the matrix *P* is expressed as follows:

$$\mathbf{y_P} = P \cdot \mathbf{x} \tag{5}$$

Finally, considering all the *SRTCs*, the following system is achieved:

$$\begin{bmatrix} \mathbf{y_L} \\ \mathbf{y_B} \\ \mathbf{y_M} \\ \mathbf{y_P} \end{bmatrix} = \begin{bmatrix} R \\ B \\ M \\ P \end{bmatrix} \cdot \mathbf{x} \tag{6}$$

Eq. (6) formalizes the *Network Status* description by means of SRv6 counters.

V. SRv6 Counters Characterization

Due to their nature, there is a strong correlation among measurements provided by the different *SRTCs*. As an example, the counters PSID(9, 6), PSID(8, 6), PSID.TM(8, 6) and POL(pol1) of Fig. 1, account exactly the same flow (f_3). Considering the model introduced in Section IV-A, this situation reflects the case of linearly dependent equations in the system of Eq. 6. The aim of this section is to identify the relationships among the measurements provided by different *SRTCs*. As it will be shown in Section VI, the knowledge of the relationships among *SRTCs* allows to improve the network monitoring operations. As a matter of example, the reduction of counters to be collected (avoiding the contemporary collection of counters values providing the same information) speed up the TM assessment procedure. Furthermore, the reduction of the *SRTCs* to be collected can mitigate the bandwidth needed to carry out the collection process. In fact, considering that the number of *SRTCs* is $\mathcal{O}(N^2)$, for big networks the collection process can requires tens of Mbps of bandwidth. By means of the proposed framework it is possible to highly reduce the bandwidth required for the collection process also in large networks (having tens of thousands of nodes), with no loss with respect to the amount of information on the *Network Status*, making a monitoring system based on *SRTCs* scalable. All the results presented in the following are obtained under the following hypothesis: i) all the nodes in the domain are SR capable,[7] ii) the central monitoring system simultaneously asks to all the nodes for the *SRTCs*, i.e., the values of the *SRTCs* are referred to the same time instant, iii) in the network none packet is dropped, and iv) sampling techniques are not applied in the update process of the *SRTCs*.

A. The Reverse Path Model

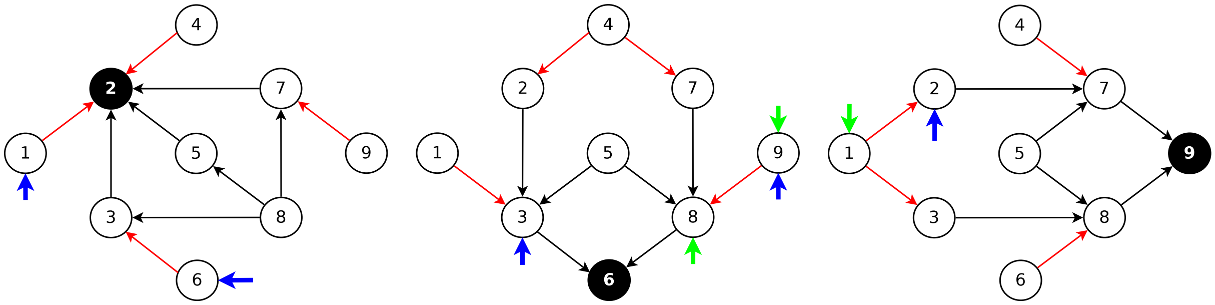
The coefficients of equations reported in Eq. 6 are strictly dependent on the routing configuration; consequently, the relationships among the counters depend on the configured routing, both at the underlay and at the overlay layers. In order to better investigate these relationships, it is useful to provide a different graphical view of the network, composed by a dedicated graph for each active segment of the network (i.e., each node of the network). More in detail, it is possible to associate to each node a its directed acyclic graphs, representing the set of reverse paths[8] from all network nodes to a. In the following we refer to a directed acyclic graph with the acronym RP. The aim of the RPs is to identify the routing in the underlay layer; in Fig. 2 the RP of a subset of nodes of the network scenario depicted in Fig. 1, are shown. Each of the RP is obtained considering the underlay paths (shortest paths computed considering that all links have the same cost). Focusing on a specific RP, e.g., the one shown in Fig. 2(a), it represents the paths followed by packets having as *actsgm* the locator of node 2; moreover, red arrows refer to external interfaces with respect to the *TM Border*.

[5] PSID(*i,i*) counter accounts all the packets for which node *i* performs the END function.

[6] It is assumed that PSID.TM counters are defined in all nodes, independently on the nodes interfaces type. If a node has no external interfaces, all the related PSID.TM are null.

[7] Anyway the results can be easily extended to the case of hybrid IP/SR domains.

[8] The RPs are not represented by trees, since IPv6 exploits ECMPs.



(a) root 2.

(b) root 6.

(c) root 9.

Fig. 2. Reverse Path Tree of different nodes of the network scenario of Fig. 1, built considering the underlay paths.

The RPs model is enriched by considering the Segment Lists and the traffic; it is worth noting that, focusing on the destination a, all the flows having an *actsgm* different than a are transparent in the related RP. On the other side, there are three different ways for a packet to have the locator a as destination:

- the node i receives the packet with *actsgm* a from a previous hop;
- the node i performs the END function and the next segment is a;
- the packet hits an *SR Policy* whose egress node has the locator a.

Analyzing the accounting rules of PSID counters and the RPs, it is possible to better investigate the previous three cases.

The first case is the easiest one, since it represents the classical behavior of a transit node, i.e., node i is only performing forwarding operation on a packet having a as *actsgm*. Considering the example of Fig. 1, this is the case of flow $f2$ at node 3. The node 3 receives the packets of the flow $f2$ having the locator of node 2 as *actsgm*. This amount of traffic is accounted in PSID(3, 2) and is also forwarded, keeping 2 as *actsgm*, to the next hop. Packets belonging to this category are referred to as *traffic accounted and forwarded*, since they are accounted at node i for *actsgm* a and they also leaves node i maintaining a as *actsgm* (i.e., they remain in the same RP).

The second case is different, since the packet is received by node i with a different *actsgm* than a. This is the case of the flow $f1$ at node 2 in Fig. 1, that performs the END function to all the packets belonging to flow $f1$ by setting the SID 9 as *actsgm*. Recalling that the PSID counters account incoming traffic, in this case, flow $f1$ is accounted by PSID(2, 2), but not by PSID(2, 9) (in fact, when the packet is received the *actsgm* is 2). Thus, if a node performs the END function by setting a as *actsgm* for an incoming packet, the PSID counter related to SID a is not updated. For this reason, we need to enhance the RP representation considering that this traffic is injected in the RP of a in node i, even if it is not accounted in counter PSID(i,a). In the RP, this situation is represented by the blues arrows, representing *traffic injected but not accounted*.

In the third case, the incoming packet is received with a as *actsgm* (and so it is accounted by counter related to a), but it changes its *actsgm* due to a policy in node i. This is the case of the flow $f3$ at node 8 in Fig. 1. In this case, before hitting the policy, packets of the flow $f3$ are destined to 6; consequently, these are accounted in the PSID(8, 6). After that, the policy is applied and the flow is steered using the specific *SID list*. It implies that, when flow $f3$ leaves node 8, its destination changes, thus the node 8 is the only one that accounts this flow in a PSID counter referred to the SID 6. To take into account this situation, the green arrows are inserted in the RP representation of Fig. 2. The green arrows show *traffic accounted but not injected*, i.e., traffic accounted in counter PSID(i,a) but leaving node i with an *actsgm* different than a (so moving to a different RP).

Starting from the previous observations and using the RP representation, it is possible to obtain the expressions describing the way PSID counters are updated; focusing on PSID counter defined at node i for the SID a, the specific expression is represented by the following equation:

$$y_B(i,a) = \sum_{l\in\delta_i} g_{l,\eta(j,a)}[y_B(j,a) - \gamma(j,a) + \beta(j,a)] \quad (7)$$

where δ_i represents the set of incoming arcs of node i, $\gamma(j,a)$ is the amount of *traffic accounted but not injected* (green arrow) at node j for SID a, and $\beta(j,a)$ is the amount of *traffic injected but not accounted* at node j for SID a. Eq. 7 shows that the value of the PSID counter at node i for the SID a depends on the traffic received from neighbor nodes. Specifically, the amount of traffic having a as active segment and forwarded to node i by its neighbor j, is given by the components: i) traffic accounted ($y_B(j,a)$); ii) traffic accounted but not forwarded ($\gamma(j,a)$), that has to be removed; and iii) traffic not accounted but forwarded ($\beta(j,a)$), that has to be added. Clearly, these contributions have to be multiplied by the coefficient of the matrix G. In particular, if node j is not using i as next hop to reach the SID a, then the related coefficient of the matrix G is equal to 0, and its contribution is set to zero in Eq. (7). An example of PSID update in accordance to Eq. 7 is given by considering the counter PSID(6, 6). As shown by the RP depicted in Fig. 2(b), node 6 is used as active segment for the destination 6 by nodes 8 and 3. Regarding node 3, it receives the flow $f3$ when it is destined to SID 3, consequently $y_B(3,6) = 0$. Anyway, node 3 performs the END function on flow $f3$, by setting 6 as *actsgm*. Then, $\beta(3,6) = f3$. At node 8, flow $f3$ is received with 6 as active segment, thus $y_B(8,6) = f3$, and the pol1 is applied to f3, i.e., $\gamma(8,6) = f3$. According to Eq.(7) $y_B(6,6) = f3$.

B. Analysis of the Relationship Between SRTCs

The RP model and the formal expression of PSID counters accounting represent the starting point to analyze the relationship among the different SRTCs. The first relationship investigated is the one among the SR.INT and the PSID counters. Considering link *l* connecting nodes *i* and *j*, the value assumed by the related SR.INT, i.e., $y_L(l)$, can be computing as the weighted sum of all the PSID counters defined at the tail node of the considered link. Specifically, the SR.INT(*i, j*) update process is described by the following equation:

$$y_L(l) = \sum_{a=1}^{N} g_{l,\eta(i,a)}[y_B(i,a) - \gamma(i,a) + \beta(i,a)] \quad (8)$$

Eq. (8) shows that, the amount of traffic sent over the link *l* can be determined by summing up the value of the PSID counters instantiated at node *i* over all the possible locators. Each of these counters must be multiplied by the $g_{l,\eta(i,a)}$ parameter, which takes into account whether or not the link *l* is contained in the IGP path between nodes *i* and *a*. Furthermore, each of these counters needs to be "modified", by including all the traffic *injected but not accounted* and by removing the traffic *accounted but not injected*. From Eq. (8) it is clear that, if the terms $\gamma(i,a)$ and $\beta(i,a)$ are all 0 (for all possible locators), then the value of the SR.INT(*i, j*) counter can be determined as a simple summation of PSIDs. In such a case, the row of the matrix *R* related to the link *l* is linearly dependent on the rows of the matrix *B* related to PSID counters instantiated at node *i*.

The second relationship investigated is the one existing among different PSID counters, which is directly related to the rank of the matrix *B*. Starting from Eq. (7), the following sufficient condition for a specific PSID counter to be linearly dependent than different ones can be derived based on the previous observations:

Theorem 1: Considering the PSID(*i, a*), IF in all the neighbours of the node *i*, considering the *actsgm a*, there is only traffic accounted and forwarded, THEN the equation written by using the measurement of the PSID(*i, a*) is linearly dependent than the rows of the matrix *B* related to neighbors of *i*.

Proof: The proof can be directly obtained by considering Eq. (7). In fact, assuming that all the neighbors of node *i* in the RP for node *a* have no green or blue arrows, then the following condition is obtained:

$$y_B(i,a) = \sum_{l \in \delta_i} g_{l,\eta(j,a)} y_B(j,a) \quad (9)$$

From Eq. (9) it is clear that the PSID(*i, a*) can be obtained as a linear combination of the counters defined in the neighbors nodes. ■

As an example, referring to Fig. 2(a), the PSID counter at node 7 can be determined by summing up the value of the counters at nodes 8 (which has to be weighted considering the ECMP) and 9. The main consequence of Theorem 1 is that the rank of matrix *B* is strictly related to the *SID lists* used, since blue and green arrows are placed in the RPs considering the SIDs of the *SID lists*.

The third relationship investigated concerns the rank of the matrix *M*, i.e., the set of equations written using the measurements achieved by PSID.TM counters. In particular, the following Theorem 2 can be proved:

Theorem 2: IF the *TM Border* is at the edge of the considered SR domain AND there is at least one traffic flow between every nodes pair, THEN the rank of the matrix *M* is maximized and equal to $N \cdot (N-1)$.

Proof: The logic of the proof consists in showing that, in the considered scenario, there are exactly $N \cdot (N-1)$ rows of the matrix *M* that are linearly independent. The proof consists of three steps.

In the first step, PSID.TM counters defined in different nodes and referred to the same *actsgm* (PSID.TM(*i, a*) and PSID.TM(*j, a*), $\forall i, j \in \mathcal{N} - a$) are considered. In the second step, PSID.TM counters defined in the same node and referred to different *actsgms* (PSID.TM(i, a_1) and PSID.TM(i, a_2), $\forall a_1, a_2 \in \mathcal{N} - i$) are evaluated. Finally, in the third step PSID.TM(*i, i*) counters ($\forall i \in \mathcal{N}$) are considered.

The first step regards PSID.TM counters of different nodes. At node *i*, PSID.TM(*i, a*) measures the overall traffic intensity of flows having node *i* as ingress point of the *TM Border* and *a* as *actsgm*. Due to the fact that the *TM Border* is at the edge, a flow is injected in the SR domain from a single ingress node.[9] As a consequence, considering two different nodes, *i* and *j*, the set of flows measured by the two counters PSID.TM(*i, a*) and PSID.TM(*j, a*) is always different and disjoint, since the considered ingress point changes (*i* and *j* respectively). This implies that the vectors $\mathbf{m}^{(\mathbf{i},\mathbf{a})}$ and $\mathbf{m}^{(\mathbf{j},\mathbf{a})}$ are orthogonal, since the non zero elements in the first vector are in different positions with respect to the non zero elements of the second vector. Consequently, the equations written considering these two counters are independents. More precisely, the previous observation allows to define *N* disjoint sets of *N* − 1 linearly independent rows of the matrix *M*. Each of these sets (referred to as M_a) is composed of all the equations written by counters that account traffic with respect to a given *actsgm*.

In the second steps, PSID.TM counters of the same node with different *actsgm* are evaluated. Considering two PSID.TM counters defined in the same node *i*, but referred to two different *actsgms*, a_1 and a_2, the set of flows they measure are disjoints. This is a consequence of the fact that a traffic flow enters the network with a single *actsgm*. This implies that the vectors $\mathbf{m}^{(i,a_1)}$ and $\mathbf{m}^{(i,a_2)}$ are orthogonal, since the non zero elements in the first vector are all in different positions with respect to the non zero elements on the second vector. Consequently, the equations written considering these two counters are independents. It also implies that, given two equations coming from two different sets M_{a_1} and M_{a_2}, they are linearly independent. Thus, it has been shown that: i) the $N-1$ equations in a set M_a are linearly independent, and that ii) all the equations belonging to two different sets (M_{a_1} and M_{a_2}) are linearly independents. Since there are *N* of these sets, it implies that exists a set of $N \cdot (N-1)$ independent equations.

[9] This is not true if the *TM Border* is inside the domain, since SR exploits ECMP.

Globally there are N^2 PSID.TM counters. Up to now it has been shown that the rank of the matrix *M* is, at least, equal to $N \cdot (N-1)$. In the next it is proven that, the remaining *N* counters do not add information. This is done by analyzing the third step. Due to the fact that the *TM Border* is at the edge of the SR domain, the first *actsgm* of a packet entering the network from an ingress node *i* can never be the SID of *i* itself. Consequently, the counters of the type PSID.TM(i, i) do not account any flow. Thus their value is always zero, and the related rows of the matrix *M* have all zero elements. Globally, there are *N* equations related to PSID.TM(i, i) counters. ■

The next investigation concerns the relationship among PSID and PSID.TM counters. To formalize this relationship, the set δ_i^{EXT} is introduced: it contains all the nodes connected to an external interfaces of node *i*. Then, the following Theorem can be proved:

Theorem 3: Considering the PSID.TM(*i, a*) counter, IF all the nodes in δ_i^{EXT}, have only traffic accounted and forwarded for *actsgm a*, THEN the equation related to PSID.TM(*i, a*) counter is linearly dependent than the rows of the matrix *B* related to nodes in δ_i^{EXT}.

Proof: Eq. (7) can be easily extended to the case of the PSID.TM counters. Specifically, considering PSID.TM(*i, a*), the update rule follows the formula:

$$y_M(i,a) = \sum_{l \in \delta_i^{\mathrm{EXT}}} g_{l,f(j,a)}[y_B(j,a) - \gamma(j,a) + \beta(j,a)] \tag{10}$$

In case the $\gamma(j,a)$ and $\beta(j,a)$ are 0, for all the nodes $j \in \delta_i^{\mathrm{EXT}}$, then the Eq. 10 shows that the PSID.TM(*i, a*) can be obtained as a linear combination of rows of the matrix *B*. ■

Finally, the contribution of the matrix *P* is studied. As first, the following Theorem is provided.

Theorem 4: Let Π_e and Π_i be the set of SR policies installed at the edge and internal nodes of the SR domain respectively, such that $\Pi = \Pi_e \bigcup \Pi_i$. THEN considering the rank of matrix *P* the following condition holds:

$$\mathrm{rank}(P) \geq |\Pi_e| \tag{11}$$

Proof: Let P_e be the matrix obtained by considering all the rows of the matrix *P* referred to SR Policies in Π_e. Since each flow hits an *SR Policy* at the edge before entering the domain, then $|\Pi_e| \leq K$ (where the equality is obtained when each *SR Policy* at the edge is used for a single flow). Now, considering the generic flow *f*, when entering the domain it hits exactly one *SR Policy*. Consequently, there is only one row of the matrix P_e having a non zero element in position *f*. By induction, each row of the matrix P_e has non zero elements in different positions with respect to the other rows (since each policy catch a different subset of flows). Then, the rank of the matrix P_e is equal to $|\Pi_e|$. Focusing on the matrix *P*, it implies that there are at least $|\Pi_e|$ linearly independent rows. ■

In the special case that each *SR Policy* in Π_e matches with a single flow, then $|\Pi_e| = K$. As a consequence of Theorem 4, the rank of the matrix *P* is equal to the number of flows.

Furthermore, with reference to the RP model, the POL counters can be used to assess the terms $\gamma(i,a)$ reported in Eq. (7), i.e., the amount of traffic accounted but not injected at node *i* for the *actsgm a*. The reason is that this type of traffic is due to the presence of SR policies installed at node *i*. Referring with $\Pi_\gamma(i,a)$ to the set of SR policies installed at node *i* and that are applied to all the traffic flows whose destination is SID *a*, then the term $\gamma(i,a)$ can be computed as follows:

$$\gamma(i,a) = \sum_{\pi_j \in \Pi_\gamma(i,a)} y_P(\pi_j) \tag{12}$$

The previously introduced Eq. (12) allows the assessment of the value, in terms of unit of traffic, of the green arrows reported in the RP model. Once their value is assessed, with reference to Eq. (7), they can be considered as known quantities, and treated as if their value were null. The rank of the matrices *B* and *M* is affected by this fact, according to the Theorems 1 and 3.

To summarize, the following outcomes from the *SRTCs* relationships characterization are obtained:

- the number of *SRTCs* measurements to detect the *Network Status* can be reduced focusing only on independent *SRTCs*, to be identified only checking the *SID lists* encoding strategy (i.e., the *SID lists* structure); the counters providing correlated measurements are SR.INT, PSIDs and PSID.TM of neighbor nodes;
- the PSID.TM counters provide the higher level of useful information about the network status if the *TM Border* is at the edge of the SRv6 domain;
- the *SR Policy* counters always provide independent equations to the *Network Status* system.

One of the main outcome of the proposed framework is that only a subset of the *SRTCs* instantiated in nodes performing the END operation are always linearly independent than the other ones. This observation, along with further ones obtained in the next section, can be exploited by network providers to reduce the complexity of the *SRTCs* collection phase, and by vendors to develop lightweight implementation of *SRTCs* on SR capable nodes, without affecting the overall amount of achieved information.

Before concluding, a discussion about the hypothesis under which the proposed framework holds is proposed. Regarding the availability of a full SR network, instead of a partially deployed one, here it is argued that the presented models can be easily extended to the case of hybrid IP/SR domains. In particular it is sufficient to remove non SR capable nodes from the RP model. Concerning the remaining hypothesis (synchronization of the measurements performed by different *SRTCs*, no packet loss and no sampling during counter update), it is clear that they do not hold in a real scenario and because of that, all the presented formulas do not strictly apply. Anyway, as reference to the first and the last one, the mismatch caused on the *SRTCs* caused by the removal of the synchronization and the no sampling hypothesis is contained. In particular, despite it is not possible to guarantee that the *SRTCs* are referred to the same exact time instant, it is possible to reduce the synchronization error.[10] Furthermore, if all nodes perform the

[10] For instance, knowing the RTT between the central controller and the nodes, the sending of counter request messages can be scheduled so that they are received almost at the same time by all the routers.

measurement adopting the same sampling rate, the discrepancy on the obtained values is restrained. As a conclusion, the present framework can be adapted to the removal of these two hypothesis by considering a safety margin and considering inequalities (instead of equations).

The situation regarding the removal of the hypothesis on the packet loss is more critical, since in a network there are many different sources of packet loss (e.g., congestion, failures, misconfigurations, packet errors, etc.). The possible mismatch between *SRTCs* due to packet loss caused by congestion of packet errors can still be included in the model, by considering a safety margin. The setup of the margin value can be driven by the availability of statistics on packet dropped (commonly available in the interfaces) and by the knowledge of the links utilization. On the contrary, the same approach cannot be successfully applied in case of failures and misconfiguration events. This last fact open the way to the identification of a different usage of the proposed framework. Specifically, the presence of packet loss makes two counters (that in normal conditions are linearly dependent) be independent. It has been argued as regular packet loss (such as congestion) can be included in the model by considering a margin. Anyway, failures or misconfiguration events can create a mismatch on the *SRTCs* that is even large than the margin. Looking at this situation it is possible to define a failure and misconfiguration detection tool. The proposition of such a framework can be the subject of future investigation.

VI. Performance Evaluation

This section is organized as follows: i) as first, some details about the simulation setup are given, then ii) a general performance evaluation is carried out, finally iii) two case studies (Traffic Matrix Assessment and Traffic Anomaly Detection) are discussed. The aim of the general evaluation is to provide an analysis of the relationships between the structure of the SID lists and the amount of information provided by different *SRTCs*. The outcomes of our analysis can be exploited in the following way: i) in a full-*SRTCs* scenario, it is possible to reduce the amount of information to be collected by the OAM focusing only on useful *SRTCs*; ii) in a partially deployment scenario, where only a limited set of routers is compliant with *SRTCs* or only a subset of *SRTCs* types are supported, it is possible to provide guidelines for the selection of routers to upgrade with partial or full *SRTCs* support.

A. Performance Model

Four real networks are selected from [11] and [12]: Abilene ($N = 12$, $L = 72$), Germany ($N = 50$, $L = 176$), RF1239 ($N = 315$, $L = 1944$) and RF1755 ($N = 87$, $L = 322$); real traffic matrices, link length and capacity are available for these networks. IGP paths are computed using the Dijkstra algorithm considering unitary link weights. The source code of the MATLAB simulator can be found in [13].

Two different approaches, referred to as *random* and *strategic*, are used to generate *SID lists*. The random strategy consists in the application of the following procedure: i) as first, a percentage $d_{\%}$ of network nodes used as middle points in the *SID lists*, is selected, ii) the network nodes are sorted with respect to their betweenness, iii) the first $d_{\%}$ nodes of the list are selected to be the SIDs in the *SID lists*, iv) the *SID list* length (SLL) is chosen and, finally, v) for each pair of IE nodes, SLL middle points are randomly extracted (no SIDs repeated in a *SID list*). In the strategic case, four different traffic classes, named colors, are considered, i.e., best effort, maximum throughput, low latency and Traffic Engineering (TE). Each Ingress-Egress (IE) traffic flow is decomposed in four sub flows (one for each color). The amount of traffic associated to each color is determined by a splitting parameter. For each color, a *SR Policy* at the ingress node is considered. Best effort traffic is routed over IGP paths, consequently, the resulting *SID list* is composed by a single SID indicating the egress node. For the other colors, the path is found, then the related *SID list* is derived using the approach described in [14]. In case of maximum throughput color, the traffic flow is sent over the path maximizing the capacity of the bottleneck link (looking at the overall bandwidth, and not the available). In case of low latency color, the path that minimizes the delay (only considering transmission and propagation) is searched. Finally, when TE color is required, then the flow is sent over the path with the minimum congestion.

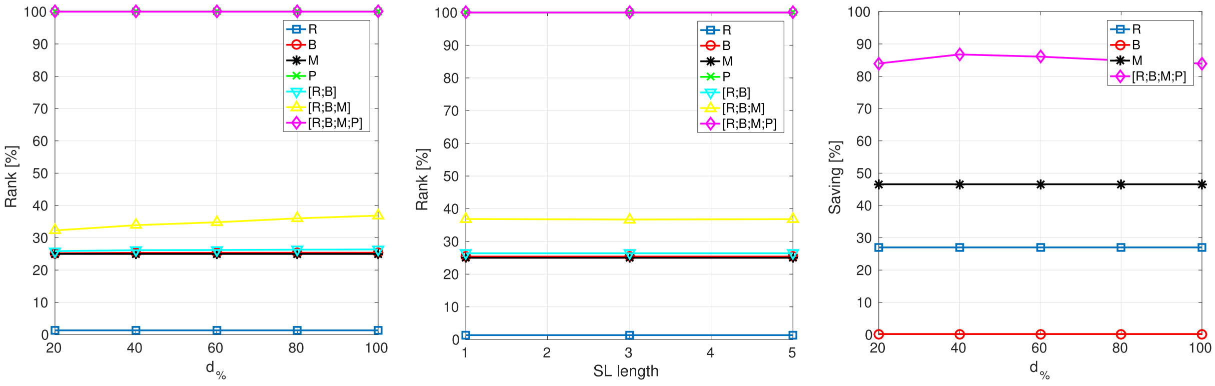
B. General Evaluation

In this section the impact of the structure of *SID lists* (in terms of SLL and $d_{\%}$) on the amount of information captured by SR counters is studied. The ratio between the rank and the overall number of flows (rank[%]) is considered as a quantitative measure of the amount of information achieved by *SRTCs*. The RF1755 topology, with *TM Border* configured at the edge of the domain and randomly generated *SID lists*, is considered as input for this analysis. Four different colors are assumed to be available. Similar results were found for the other networks.

The rank of different matrices as a function of the $d_{\%}$ parameter is shown in Fig. 3(a). In this case the SLL is fixed to 2, which means that the path between every couple of ingress and egress nodes is composed of two segments (from the ingress to middle point and, from the middle point to the egress). For the ease of comprehension, before commenting the results of the conducted analyses, it is recalled that SR.INTs, PSIDs, PSID.TMs and POLs are related to R, B, M and P matrices, respectively.

A first expected outcome of the analysis reported in Fig. 3(a) is that PSIDs, PSID.TMs and POLs provide more information than SR.INTs; in fact, the rank of the matrices B, M and P is always greater than the one of matrix R. This finding is of interest since, SR.INT counters provide a measurement similar to the one normally available in all the network interfaces, i.e., the amount of data transmitted (link count). Then, PSIDs, PSID.TMs and POLs allow to improve the knowledge of the *Network Status* with respect to classical link count approaches.

A second aspect to be highlighted is that PSID and PSID.TM counters provide a similar level of information on the *Network Status*. The reason is that, taking into account that in the considered network the *SR Policy* are installed in



(a) Rank VS $d_{\%}$.

(b) Rank VS *SID list* length.

(c) Saving VS $d_{\%}$.

Fig. 3. Analysis of the information achieved by *SRTCs* with respect to *SID list* structure.

every node (all nodes are at the edge of the considered SR domain), all the rows of the *B* matrix are independent, as a consequence of Theorem 1. Moreover, according to the theorem 2, the information provided by PSID.TM counters is independent on the value of $d_{\%}$. In fact, due to the way this type of counters work and to the fact that the TM border is set at the edge of the considered SR domain, PSID.TMs are incremented before that the *SID list* is attached to the packets. Specifically, it is interesting to notice that they introduce an amount of information equal to 25%. Considering that (in the simulation setup) there are 4 traffic flows between each pair of nodes, the overall number of flows is $K = 4 \cdot N \cdot (N-1)$. Then, it implies that the number of linearly independent equations in the matrix *M* is given by $0.25 \cdot K$, i.e., $N \cdot (N-1)$, which is in line with Teo. 2. Focusing on PSID counters together with PSID.TM and SR.INT counters (matrix [*R; B; M*]), it can be seen that the amount of the information they provide depends on the $d_{\%}$ parameter. Specifically, the rank[%] increases according to the cardinality of the set of locators that can be used as middle points. This is a direct consequence of the Theorem 1. In fact, when the $d_{\%}$ parameter grows, the number of blue arrows, defined in RP model presented in Section V, increases, weakening the linear dependence of the rows of the *B* matrix.[11] At the same time, the results obtained highlight that the rank of the matrix [*R; B; M*] is much lower than the sum of the ranks of the single matrices *B* and *M*: this means that many rows of the matrices *B* and *M* are linearly dependent, as stated by Theorem 3. In particular, in the considered simulation setup each PSID.TM counter is able to measure the *traffic accounted but not injected* ($\gamma(j, a)$ terms in Eq. 10). As a consequence, the only terms that allows to PSID and PSID.TM counters to be independent are the ones related to the *traffic injected but not accounted*. These lasts increase in number as the $d_{\%}$ parameter grows (more END functions are applied). A last important remark that comes out from the analysis reported in Fig. 3(a), is that, by considering SR.INTs, PSIDs and PSID.TMs it is possible to get only a partial information about the *Network Status* (observe that the rank of the matrix [*R; B; M*] is close to 40% when $d_{\%} = 100\%$). This gap can be easily filled by considering the information achieved by POL counters. Specifically, these lasts alone achieve the full knowledge about the *Network Status*, in fact the rank of *P* matrix is always 100%, i.e., the two lines referred to *P* and [*R; B; M; P*] cases in Fig. 3(a), are overlapped.

The evaluation of the impact of the *SID list* length on the information achieved by *SRTCs* is reported in Fig. 3(b). For this analysis, the value of $d_{\%}$ parameter is set to 100%. By inspecting Fig. 3(b) it is evident that the rank[%] of all the *SRTCs* related matrices does not depend on the *SID list* lengths. An explanation for this finding can be obtained by Eq. 7, that shows which are the parameters that influence the update of a PSID counter. In particular, the values of $\beta(j, a)$ and $\gamma(j, a)$ depend on $d_{\%}$ and the number of SR policies respectively, thus they do not strictly depends from the *SID list* length.

In order to quantify the benefit achieved by the various theorems presented in Section V, in Fig. 3(c) an analysis of the saving, intended as the ratio between the number of identified useless *SRTCs* (i.e., linearly dependent) and the overall number of counters, as a function of $d_{\%}$ parameter is reported. Two main outcomes are obtained: i) exploiting the results of the presented theorems, it is possible to significantly reduce the dimension of the matrices related to *SRTCs*, and ii) the presented theorems represent only sufficient conditions for linear dependence. This last point can be seen in Fig. 3(c), by referring to the line showing the saving for the full set of *SRTCs* related matrices. As it can be seen, the saving is not constant, but presents small variations. At the same time, the rank keeps constant (see Fig. 3(a)). It means that the overall matrix has a higher number of rows, but the same rank; it implies that some of them are linearly dependent.

The impact of the configuration of the *TM Border* on the information achieved by *SRTCs* is now analyzed. Specifically, only PSID.TM counters are considered (which are the only ones affected by this configuration). This evaluation is carried out considering strategic *SID lists*. The *TM Border* is iterative changed as follows: i) nodes are sorted according to a given criterion; ii) at each iteration a node is inserted

[11] Considering Eq. 7, the $d_{\%}$ affects the number of $\beta(j, a)$ terms that are different than 0.

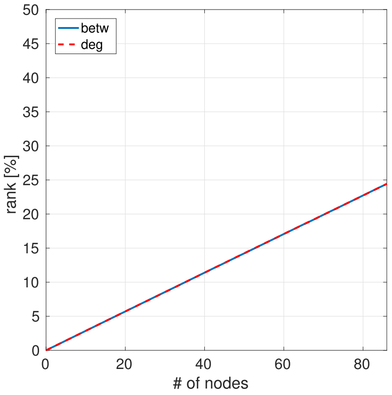


Fig. 4. Impact of the *TM Border* on the information provided by PSID.TM counters.

into the *TM Border*; iii) the connectivity of the obtained *TM Border* is checked; iv) all the interfaces between nodes into the *TM Border* and other nodes are marked as external. Two different parameters are used to accomplish point one, namely betweenness and degree. More precisely, the betweenness of a node is equal to the number of shortest paths that contain the node, while the degree is represented by the number of links connected to a node. The two different criteria lead to two different *TM Border* (even if the number of nodes of the *TM Border* is the same). The results of the proposed analysis are reported in Fig. 4. It is clear that, as soon as the *TM Border* gets closer to the edge of the considered SR domain, the amount of information achieved by the PSID.TM counters increases. Moreover, it is evident that the best choice (whenever possible) is to set the *TM Border* at the edge of the SR domain, since it leads to the highest level of information. Further, if the *TM Border* does not coincide with the SR domain, the design of the *TM Border* does not seem to significantly affect the overall information achieved. Specifically, *TM Border* of the same size (in terms of number of nodes) achieve comparable information.

Summarizing, the insights arising from the previous analysis are: i) the amount of information on the *Network Status* achieved by the use of *SRTCs* depends on the *SID lists* structure, ii) *SRTCs* allow to significantly improve the knowledge of the *Network Status* with respect to the simple link counts, iii) POL counters allow the complete knowledge of the *Network Status*, and iv) concerning the *TM Border* configuration, the best option is to have it at the edge of the considered SR domain.

C. Case Study 1: Traffic Matrix Assessment in a Partially Deployed Scenario

The Traffic Matrix (TM) of a network is a data structure containing the volumes of the traffic flows exchanged between nodes. Depending on the level of detail, different TMs can be defined, each one useful to accomplish a specific task (TE, network design, etc.). A classical representation of the TM is the Ingress-Egress Traffic matrix (IE-TM), where each element of the TM is the volume of traffic flowing between a couple of ingress-egress nodes. Another classical representation of the TM is the Origin-Destination (OD-TM) one. In particular a single IE traffic flow is the aggregation of many OD flows.

A good knowledge of the TM is a crucial requirement for Network Operators. For instance, IE-TM can be used to accomplish network design, capacity planning operations, etc. [15]. Despite many works are focused on TM Assessment (TMA) [4], it is still considered an open issue.

In the next a qualitative and quantitative analysis of the impact of *SRTCs* availability in the TMA problem is provided. As first it is argued how, by using the *SRTCs*, the TM can be easily measured. Then, a performance evaluation is conducted to determine the quality of the assessed TM when a partial deployment scenario (only a subset of nodes implement *SRTCs*) is considered.

The classical formulation of the TMA problem consists in the assessment of the amount of traffic exchanged between each pair of nodes in the network, only relying on the knowledge of the routing and the amount of traffic over each link. The mathematical model to represent the TMA is similar to the one that describes the *Network Status*, and given by the formula reported in Eq. 2. It consists of a system of linear equations, where instead of the vector $\mathbf{x}$ of the intensity of the traffic flows steered through the SR Policies, there is the vector $\mathbf{x}_{\mathrm{IE}}$ of the unknown intensities of the IE traffic flows. Both the vector $\mathbf{y}_{\mathbf{L}}$ and the matrix R assume the same meaning of the ones reported in Eq. 2. In particular, the following Eq. 13 expresses the relation between an IE traffic flow and SR traffic flows (elements of the vector $\mathbf{x}$):

$$x_{\mathrm{IE}}^{i,e} = \sum_{\forall k: f(k)==(I,E)} x_k \tag{13}$$

where $f : 1 \dots K \rightarrow \mathcal{N} \times \mathcal{N}$ is a function that maps the flow ID with the corresponding IE pair. Eq. 13 states that, the intensity of the IE flow between nodes i and e is computed by summing up the intensities of the flows that enter the SR domain from the node i and leave it through the node e.

Unfortunately, the resulting linear system is highly under determined in real network scenarios, since in general $L << N \cdot (N-1)$ (this last quantity represents the number of nodes pairs). As a consequence, the system cannot be inverted and there are infinite values of the vector $\mathbf{x}_{\mathrm{IE}}$ that satisfy Eq. (2).

The classical TMA can be enhanced by considering also the set of measurements achieved by *SRTCs*. Specifically, once all the *SRTCs* are considered, the linear system shown in Eq. 6 is obtained. Now, since in the analysis reported in Fig. 3(a) it is shown that the overall matrix describing the linear system reported in Eq. 6 has full rank, consequently it can be inverted to get the vector $\mathbf{x}$. After that, the TM can easily be got by applying Eq. 13.

A further interesting aspect regards the use of PSID.TM counters for IE-TM assessment. Specifically, it is possible to prove that when the *TM Border* is the edge of the SR domain, then PSID.TM counters directly measure the IE-TM (without requiring an estimation phase). In fact, according to the description of the update process of PSID.TM counters provided in Section III, under this specific configuration, each packet entering the SR domain is accounted with respect to its egress node. This is because, before entering the SR domain, packets are processed using an *SR Policy* at the ingress point. In particular, the ingress node receives the packet (from an external interface), inspects the *SR Policy* table and process the packet accordingly. Furthermore, it also gets the locator of the egress node, which is used to update the related PSID.TM counter. As a consequence, considering ingress node i and egress node e, the PSID.TM(i, e) is accounting all the traffic injected in the SR domain from node i and leaving it through node e, which corresponds to an IE flow. Then, if the *TM Border* is at the edge of the SR domain and the target is the IE-TM, PSID.TM counters can be exploited in place of all the other *SRTCs*, thus reducing the information to collect.

Starting from the last consideration, in the following it is proposed an evaluation of the level of details that can be reached in the knowledge of the TM, when only a sub set of *SRTCs* is available. In particular, from now on the target is not the IE-TM but its Origin-Destination version (OD-TM). The aim of this analysis is twofold: i) the determination of the level of performance achieved in case of a hybrid IP/SR domain [16], and ii) the evaluation of the case of *SRTCs* partial implementation, i.e., only a subset of counters type is actually available. The analysis is carried out over the Germany topology, considering strategic *SID lists* and the *TM Border* being placed at the edge of the considered SR domain. Since from the evaluation reported in Section VI-B it is evident that in partial deployment cases the full OD-TM cannot be assessed, an estimation algorithm is applied to produce a final outcome. Specifically, the Tomogravity estimator [17] is used. The estimation error is measured by means of the Relative Root Mean Squared Error (RRMSE) parameter. Before commenting the results it is worth to mention that the acceptable value of RRMSE on the estimated TM depends on the targeted application. For instance, if the estimated TM is used as input for a Network Capacity Planning application, then larger value of RRMSE (e.g., 0.4) can be tolerated. On the contrary, for applications that requires Online Routing Optimization, more accurate estimations have to be considered (e.g., RRMSE lower than 0.2).

The results of the first analysis conducted over the Germany network are shown in Fig. 5, where the RRMSE of the OD-TM estimation is reported when only a subset of *SRTCs* types is available in all network nodes. As first, it is important to highlight the great advantage achieved by the introduction of the *SRTCs* in the TMA problem: the improvement in the estimation when PSID, PSID.TM and POL counters are available with respect to SR.INT counters (which represents the link load) is significant. Focusing on PSID counters, the average RRMSE with respect to the simple SR.INT count information is about 4 times smaller. Furthermore, as expected, when only PSID.TM or POL counters are available, the OD-TM can be assessed with no error. In particular, the result obtained for the POL counters is a direct consequence of the Theorem 4: in the considered scenario the rank of the matrix P is equal to the number of traffic flows, thus allowing the direct measurement of the TM. Similarly, the result obtained for the PSID.TM is a direct consequence of the application of Theorem 2, that in the considered scenario allows the direct measurements of the IE traffic flows. The main outcome of this first study is that, from the TMA perspective, the number of needed counters can be drastically reduced, i.e., focusing only on PSID.TM ones. This improvement impacts both routers (no need to update and store many different counters) and the management system, (no need to collect the measures of many different counters).

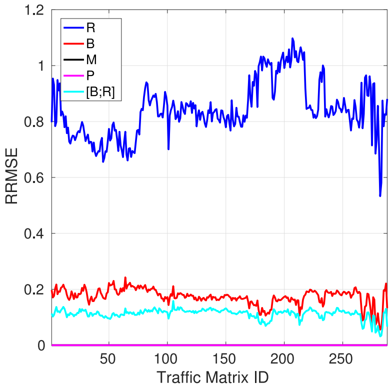
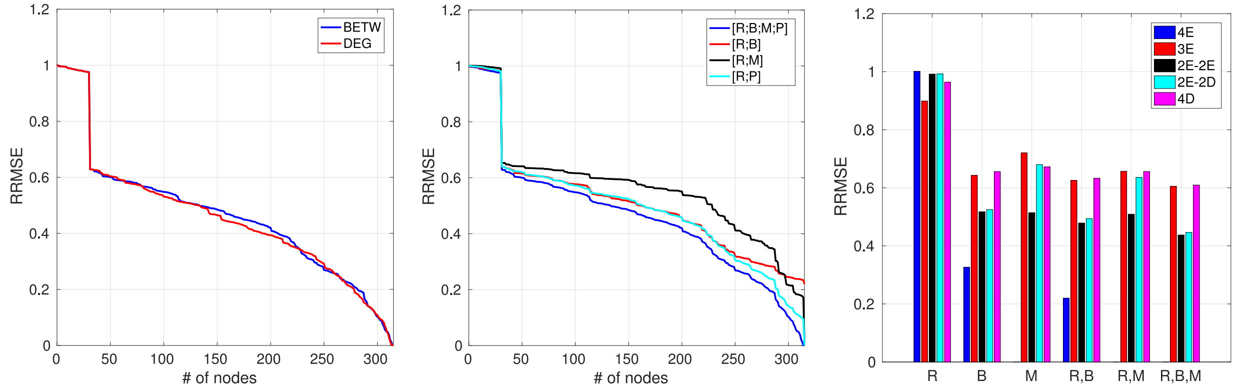


Fig. 5. Estimation Error over time considering different types of *SRTCs*, where the relation between matrices and *SRTCs* is as follows: i) R - SR.INT, ii) B - PSID, iii) M - PSID.TM, iv) P - POL.

To better investigate the possibility of reducing the types of implemented *SRTCs*, Fig. 6 reports the evaluation of a hybrid scenario, where a subset of types of *SRTCs* is available over a subset of nodes. Specifically, Fig. 6(a) reports the RRMSE as a function of the number of nodes implementing all *SRTCs*. The results are referred to the RF1239 network. The nodes are selected according to two different strategies, i.e., in descending order of betweenness (i.e., number of shortest paths containing the node) or degree (i.e., number of links connected to the node). As expected, as the number of nodes with available *SRTCs* increases, the RRMSE decreases. The result shown in Fig. 6(a) suggests that in big networks the estimation error can be reduced simply increasing the number of nodes implementing the *SRTCs*, regardless of the specific nodes selected. Anyway, it is important to mention that in smaller networks different set of nodes implementing *SRTCs* provides highly different values of estimation error, with the betweenness based method allowing for better performance than the degree based one. Finally, it is evident that the estimation error becomes negligible only when almost all the network nodes have *SRTCs* implemented.

In the next analysis (Fig. 6(b)) the partial implementation of the *SRTCs* with respect to the set of nodes and the types of counters, is considered. In this case, nodes are sorted according to the betweenness parameter. The baseline is represented by the blue curve, which reflects the case where all the types of *SRTCs* are implemented on a subset of nodes. All the other lines are referred to a specific type of counter. Here it is assumed that SR.INT counters are always implemented, since these lasts are similar to classical link counts that are commonly available in every line card. The result reported in Fig. 6(b) shows that, when only a subset of nodes are *SRTCs* available, the highest information is achieved by PSID and POL counters, while PSID.TM get higher RRMSE values. This result highlights an important aspect to be taken



(a) Subset of nodes implementing all *SRTCs*. (b) Subset of nodes implementing a subset of *SRTCs*. (c) Estimation Error in partially deployed scenario for different splitting ratios.

Fig. 6. Evaluation of the estimation error over a partially deployed scenario, where the relation between matrices and *SRTCs* is as follows: i) R - SR.INT, ii) B - PSID, iii) M - PSID.TM, iv) P - POL.

into account when the partial deployment scenario is considered, i.e., robustness. Thus, despite the analysis reported in Fig. 5 has shown that PSID.TM counters allow for the exact OD-TM assessment, the result obtained in Fig. 6(b) suggests that PSID.TMs are suitable only in case of a full deployment. In other cases, they have worse performance with respect to PSID counters, that are more robust in a partial deployment scenario.

To further provide an insight about the importance of the robustness of *SRTCs* information, in Fig. 6(c) the RRMSE in case a subset of *SRTCs* types is available in all network nodes, for different distributions of the *splitting ratios*, is reported. The splitting ratio $\sigma_{i,e}^{c}$ is a parameter introduced in the simulation that allows to determine the fraction of the IE traffic flow between nodes i and e associated to color c. In all the previous analyses, a flow is equally splitted among colors.[12] In particular, remembering that 4 different colours are available, the following different settings are considered in the analysis reported in Fig. 6(c): i) 4E, where the splitting ratios are all equals, ii) 3E, meaning that three splitting ratios are equal and one is different, iii) 2E-2E, where there are two pairs of equal splitting ratios, iv) 2E-2D, in which only two splitting ratios are equal, while the others are different, and v) 4D, where all the splitting ratios are different each others.

From the result reported in Fig. 6(c) it emerges that the PSID counters offer a higher robustness also with respect to the splitting ratio, when compared with different *SRTCs* types. In fact, despite the solutions considering the PSID.TM counters allow for a better RRMSE in the 4E case, in all other configurations of splitting ratios the estimation error of the plain PSID case is smaller than in all the other cases. This result can be explained considering that each PSID counter provides an higher number of conditions associated to each flow; in the case of PSID.TM, each flow impact only a single counter. The availability of more conditions on the same flow helps the estimation algorithm in improving the quality of the assessed volume.

To conclude, this analysis has shown that in a partial deployment scenario the PSID counters turn to be more reliable than the other types of *SRTCs*. This result could be considered as a guideline for vendors to select the *SRTCs* types to implement in case of a lightweight distribution of SR capable nodes.

D. Case Study 2: Traffic Flow Anomaly Detection

In this case study, the impact of the availability of *SRTCs* on the detection of anomalous traffic flows is evaluated; such flows are characterized by sudden variations (positive or negative) of their traffic volume. Anomalous behavior of a traffic flow can be seen as a security threat [18] for an ISP network, which can lead to quality of service degradation due to congestion or service disruption.

Traffic anomaly detection methods can be classified into two main categories, on the basis of information needed as input. The first category is named *single flow measurement* and requires the availability of information at flow level; specifically, the input for these anomaly detection algorithms is a time series of the volume of a single traffic flow. Some examples of approaches belonging to this category are: exponential weighted moving average (EWMA) [19], Holt-Winters forecasting algorithms [20], wavelet and Fourier analysis [21]. The second category is named *aggregated flows measurement* and allows the use of a higher level of information with respect to the methods belonging to the first category. An example of an approach falling into this class is the Principal Component Analysis (PCA) based anomaly detection [22], that requires as input a time series of link counts.

Generally, *single flow measurement* methods require the use of monitoring tools (e.g., Netflow), that use packet sampling techniques to reduce the size of the final traffic traces. Unfortunately, sampling techniques can negatively affect the precision of anomaly detection algorithms [23]. Since Theorem 4 shows that the rank of the matrix P is equal to the number of traffic flows, using POL counters allows for the exact measurement of each Origin-Destination flow. Thus, a first advantage obtained by the availability of *SRTCs* is represented by the possibility of having flow level time series, by

[12] The summation of the splitting ratios over all possible colors (keeping ingress and egress nodes constants), must be equal to 1.

means of POL counters, that are not affected by the problems correlated to sampling techniques.

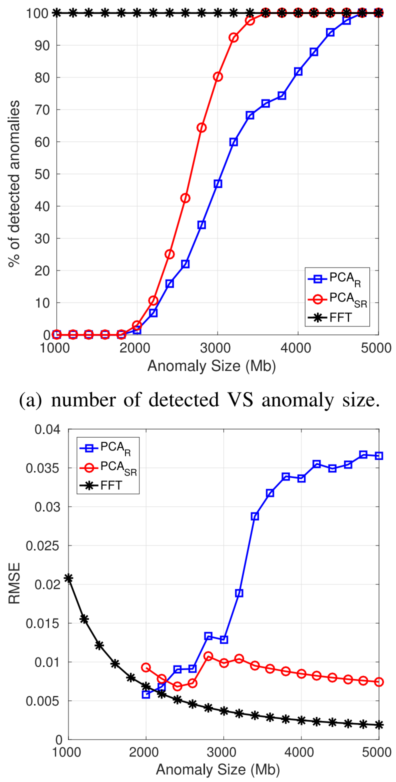
In the next, a quantitative analysis aiming at evaluating the potential benefit due to the availability of *SRTCs* on anomaly detection methods is proposed. Due to the absence of a labeled data set,[13] the performance evaluation has been carried out using synthetic traffic traces generated according to the following formula:

$$f_{i,e}^{c}(t) = \sigma_{i,e}^{c} \cdot \Gamma_G(i,e) \cdot \left(0.5 + \sin\left(\frac{2\pi t}{T}\right)\right) + n(P_N) \tag{14}$$

where, $f_{i,e}^{c}(t)$ represents the volume at time *t* of the traffic flow that cross the network from ingress node *i* to egress node *e*, and requiring a color *c*. The term $\sigma_{i,e}^{c}$ is the traffic splitting parameter, specifying the percentage of the total amount of traffic exchanged between nodes *i* and *e*, that requires color *c*, while $\Gamma_G(i,e)$ represents the average value of the total amount of traffic injected in the network from node *i* and leaving it through node *e*. This quantity is calculated according to a gravity model [17], which imposes each traffic flow to be inversely proportional to the product of the degree of the ingress and egress nodes. Furthermore, each traffic flow is assumed to have a sinusoidal behavior over time, with a period *T*. Finally, a noise term $(n(P_N))$, uniformly chosen in the interval $[-P_N, +P_N]$, is considered.

The analysis is carried out on the Abilene network, by generating three days of traffic traces. Each flow has a period of one day, and it is assumed that data arisen from *SRTCs* are collected every 10 minutes (referred to as time slot, TS), consequently, each flow is represented by a time series of 432 values. It is assumed that in a TS there is, at most, a single anomalous flow. An anomaly is generated according to the following steps: i) the sign (positive or negative) is randomly chosen, and ii) the intensity is set equal to P_A.

Three different methods are considered, named, FFT, $\mathrm{PCA_R}$ and $\mathrm{PCA_{SR}}$ respectively. FFT consists in the application of the Fourier analysis to the time series of POL counters for each traffic flow. This method divides the time series into two contributions: i) the normal signal (represented by the 8 largest Fourier basis functions), and ii) the anomalous signal. The flow is considered to be affected by an anomaly if the difference between anomalous part (in a given time instant) and the average value is greater than three times the standard deviation. PCA based anomaly detection method considers aggregated measurements as input. Specifically, $\mathrm{PCA_R}$ uses a set of time series representing the value of the SR.INT counters over the network links,whereas $\mathrm{PCA_{SR}}$ is based on a set of time series representing the value of PISD and PSID.TM counters. According to PCA general principles, the aim is to divide the time series into a regular and an anomalous part. The principal components should capture the regular contribution of the input data set (named regular subspace). A threshold-based method is exploited: the components whose values are greater than three times the standard deviation with respect to the average are marked as representing the anomalous subspace.



(a) number of detected VS anomaly size.

(b) RRMSE VS anomaly size.

Fig. 7. Application of *SRTCs* for detection of anomalous flows in the Abilene network.

Once that the regular and anomalous sub spaces have been defined, each time series is analyzed to determine whether it contains or not an anomaly. This process allows to identify the TS affected by the anomaly, but is not able to detect the anomalous flow. The procedure used to determine and quantify the anomalous flow (and quantify it) are presented in [22].

In all the reported analysis, the curves referring to the method $\mathrm{PCA_R}$ can be considered as a benchmark. This is because, the information they provide is similar to the one obtained using link counters.[14] These lines correspond then to the results achieved without the availability of *SRTCs*.

The first analysis proposed is the evaluation of the number of detected anomalies as a function of the anomaly size. For the anomaly generation, all the permutations of the 2-tuple constituted by time slot and flow ID are considered (assuming a single anomaly per TS). The noise level is fixed to $P_N = 400$Mb, and strategic *SID lists* are considered. For each anomaly value, the TS is fixed and the number of detected anomalous flows is found, then this last is averaged over all possible TSs. Results are reported in Fig. 7(a). The advantage achieved by the availability of *SRTCs* information is evident. As first (as previously discussed), *SRTCs* enable the use of the FFT method, which is always able to detect the anomalous flow; moreover it also boosts the performance of the PCA based method, by increasing its detection ability.

[13] A set of real traffic matrices with known anomalies.

[14] It's not the same since SR.INT counters take into account only SR packets while the classical link load values account any kind of traffic.

Furthermore, the use of *SRTCs* has a beneficial impact in the estimation of the anomaly size. This result is reported in Fig. 7(b), showing the RRMSE as a function of the anomaly size. As above, it is evident the improvement allowed by the use of *SRTCs* information in the anomaly detection methods (the line referred to methods based on *SRTCs* are below the result of $\mathrm{PCA_R}$). Looking at Fig. 7(b), it can be seen that for too low values of the anomaly size, the RRMSE for the PCA based methods is not defined. This is due to the fact that, in these conditions, these methods do not detect any anomalous flow. Then, it is interesting to notice that, the accuracy of the quantification obtained in case of $\mathrm{PCA_R}$ decreases (the RRMSE increases) as the size of the anomaly increases. This is due to the fact that, as the size of the anomaly grows, an higher number of them is detected. Consequently, there are more anomalies to be estimated. In case of $\mathrm{PCA_R}$ this leads to a decrease in the accuracy. On the other hand, by using *SRTCs*, due to their thinner granularity in the measurements, the quality of the quantification of the anomalies increases.

VII. Related Work

Segment Routing Traffic Counters (*SRTCs*) have been introduced in [3], with the aim of defining new solutions for Operations, Administration, and Maintenance (OAM) tasks in SR networks. This section details the state of the art about SRv6 OAM frameworks proposed in literature. A comprehensive survey of the research activities on SR is provided in [24].

The main building blocks to realize OAM in an SRv6 network are defined in [25]. This draft, firstly introduces the concept of *O Flag* (a special bit defined in the SRH), and two different functional SIDs (*END.OP* and *END.OTP*). Then, the draft explain how classical ping and traceroute applications can be performed in an SRv6 network, by means of the aforementioned building blocks. How Seamless Bidirectional Forwarding Detection (S-BFD) can be used to monitor the health of an SR tunnel is explained in [26].

A solution to measure the end to end delay based on SR is presented in [27]. The proposed framework sends packet probes along alternative end to end paths, enforced by means of different *SID lists*. The feature of SR of keeping flow state information only at the head end node of a policy, is a crucial aspect for the reliability of the proposed delay measurement tool. In [28], eBPF is exploited as a method to add programmability to the data plane, i.e., to define programs and routines to be applied on incoming packets. In particular, a new type of function SID is defined, named *END.BPF*, which allows to the source node to ask to a given network node to apply a eBPF program on a packet. Based on this, a tool to measure the one-way delay between two nodes is presented.

Detection and localization of link failures is the main target of SCMon [29]. The framework is based on a set of packet probes, which are sent over cyclic paths enforced through the use of different *SID lists*. Authors evaluate the time to detect a failure showing that, most of them are detected within less than 100 msec. A similar approach is exploited in [30], which proposes algorithms aiming at preserving the network bandwidth while probing the cyclic paths.

A different aspect of OAM is the measurement and monitoring of traffic flows. This is the main topic of the solutions proposed in [4], [31], [32], where the main goal is the assessment of the Traffic Matrix of an ISP network. Specifically, [32] proposes a framework named SERPENT, which exploit the great routing flexibility and the source routing paradigm to measure ingress-egress traffic flows. The idea is to re-route a flow and get the value of its intensity by looking at the load variation caused in the links of the new path. An ILP formulation is provided, and the problem is proven to be NP-hard. A heuristic algorithm that tries to minimize the number of rerouting operation needed to assess the full TM is proposed in [4].

Finally, the impact of the availability of *SRTCs* on the Traffic Matrix Assessment problem is evaluated in [31]. The main novelty of the present work with respect to [31] are: i) SR.INT and POL counters are considered, ii) a more realistic model is considered, where many flows can be exchanged between the same pair of Ingress-Egress nodes, iii) a theoretical framework is presented, to show the relation arising between different counters, iv) the problem is modeled in a more general way, by introducing the concept of *Network Status*, and v) an extensive performance evaluation is proposed, considering two important case studies.

VIII. Conclusion

In this paper we investigated the improvements provided by *SRTCs* availability to the monitoring phase of a Segment Routing ready ISP network. We provide a detailed review of all *SRTCs* type, defining a theoretical framework able to characterize the *Network Status* using *SRTCs* and to identify relationships among them. We also proposed two different use cases for the exploitation of *SRTCs*: i) Traffic Matrix Assessment (TMA) and ii) Traffic Anomaly Detection.

The main contribution of the paper is the detection of simple rules for the identification of useless counters: the conditions to be checked are only related to the Segment Lists structure and then can be easily computed offline. As first, we proved that it is possible to reduce the number of *SRTCs* to be collected (in a full-*SRTCs* scenario) or implemented (when upgrade of a partial deployment) on the basis of the following outcomes: i) PSID counters are independent (and so provides extra information with respect to different *SRTCs*) only in nodes performing the END function, while can be neglected in transit nodes; ii) a strong correlation exists between the information achieved by PSID and PSID.TM counters, thus only one of them can be considered. Regarding the TMA case, we proved that: i) a reduced set of counters, i.e., PSID.TM or POL one, allows to assess the full TM if all nodes are SR capable with a full *SRTCs* support; ii) the PSID counters are more reliable than the PSID.TM ones with respect to the specific Traffic Engineering policy (as a consequence, in the case of a lightweight implementation of *SRTCs*, PSID counters support should be preferred with respect to PSID.TM one); iii) the best configuration of the *TM Border* is at the edge of the considered SR domain. In the Traffic Anomaly Detection case, the following outcomes have been obtained: i) a flow based solution can be used with no need of flow level monitoring protocols, since POL counters already provide flow level information; ii) the PCA solution can use the *SRTCs* measurements both to increase the percentage of anomalies detected and to reduce the minimum anomaly size detectable, thanks to their thinner granularity with respect to link load values.

The proposed framework has also highlighted two future research directions.The first one regards the possibility of using the proposed model to detect network failures: in the case of *SRTCs* values violating the model equations, it is possible to investigate link/node failures or misconfigurations. A further research activity is the definition of novel monitoring algorithms (such as new TM estimator or new Flow Anomaly Detection algorithms) that supports and exploits the *SRTCs*.

References

[1] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Sep. 2017.

[2] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for openflow networks," in *Proc. 11th Int. Conf. Passive Active Meas.*, 2010, pp. 201–210.

[3] C. Filsfils, Z. Ali, M. Horneffer, D. Voyer, M. Durrani, and R. Raszuk, "Segment routing traffic accounting counters," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, Jun. 2018.

[4] M. Polverini, A. Cianfrani, and M. Listanti, "Interface counters in segment routing v6: A powerful instrument for traffic matrix assessment," in *Proc. 9th Int. Conf. Netw. Future (NOF)*, Nov. 2018, pp. 76–82.

[5] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture," Internet Eng. Task Force, Fremont, CA, USA, RFC 8402, Jul. 2018.

[6] A. Bashandy, C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing with MPLS data plane," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, May 2019.

[7] C. Filsfils, D. Dukes, S. Previdi, J. Leddy, S. Matsushima, and D. Voyer, "IPv6 segment routing header (SRH)," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, Jun. 2019.

[8] C. Filsfils, D. Dukes, S. Previdi, J. Leddy, S. Matsushima, and D. Voyer, "SRv6 network programming," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, Apr. 2019.

[9] C. Srinivasan, T. Nadeau, and A. Viswanathan, "Multiprotocol label switching (MPLS) traffic engineering (TE) management information base (MIB)," Internet Eng. Task Force, Fremont, CA,USA, RFC 3812, Jun. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3812.txt

[10] *Configure Segment Routing Over IPv6 (SRv6)*, Cisco Syst., San Jose, CA, USA, 2019. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-6/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-66x/b-segment-routing-cg-asr9000-66x_chapter_011.html

[11] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–survivable network design library," in *Proc. 3rd Int. Netw. Optim. Conf. (INOC 2007)*, Spa, Belgium, Apr. 2007.

[12] *DEFO Dataset*, UCLouvain, Ottignies-Louvain-la-Neuve, Belgium, 2015. [Online]. Available: https://sites.uclouvain.be/defo/

[13] P. M., (2019). *SRTC Simulator Source Code*. [Online]. Available: https://drive.google.com/file/d/1LecmAxls4pDOK6jt8nCQdzYwvduycmaJ/view?usp=sharing

[14] A. Cianfrani, M. Listanti, and M. Polverini, "Translating traffic engineering outcome into segment routing paths: The encoding problem," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2016, pp. 245–250.

[15] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Amsterdam, The Netherlands: Elsevier, 2004.

[16] A. Cianfrani, M. Listanti, and M. Polverini, "Incremental deployment of segment routing into an ISP network: A traffic engineering perspective," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3146–3160, Oct. 2017.

[17] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 206–217, 2003.

[18] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 217–228, 2005.

[19] J. D. Brutlag, "Aberrant behavior detection in time series for network monitoring," in *Proc. 14th USENIX Conf. Syst. Admin. (LISA)*, vol. 14, 2000, pp. 139–146.

[20] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: Methods, evaluation, and applications," in *Proc. 3rd ACM SIGCOMM Conf. Internet Meas.*, 2003, pp. 234–247.

[21] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, 2002, pp. 71–82.

[22] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, 2004.

[23] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of packet sampling on anomaly detection metrics," in *Proc. 6th ACM SIGCOMM Conf. Internet Meas.*, 2006, pp. 159–164.

[24] P. L. Ventre *et al.*, "Segment routing: A comprehensive survey of research activities, standardization efforts and implementation results," 2019. [Online]. Available: arXiv:1904.03471.

[25] Z. Ali, C. Filsfils, S. Matsushima, D. Voyer, and M. Chen, "Operations, administration, and maintenance (OAM) in segment routing networks with IPv6 data plane (SRv6)," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, Aug. 2019.

[26] Z. Ali, C. Filsfils, S. Matsushima, D. Voyer, and M. Chen, "Bidirectional forwarding detection (BFD) for segment routing policies for traffic engineering," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft, May 2019.

[27] F. Paolucci *et al.*, "Interoperable multi-domain delay-aware provisioning using segment routing monitoring and BGP-LS advertisement," in *Proc. 42nd Eur. Conf. Opt. Commun. (ECOC 2016)*. 2016, pp. 1–3.

[28] M. Xhonneux, F. Duchene, and O. Bonaventure, "Leveraging eBPF for programmable network functions with IPv6 segment routing," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.*, 2018, pp. 67–72.

[29] F. Aubry, D. Lebrun, S. Vissicchio, M. T. Khong, Y. Deville, and O. Bonaventure, "SCMoN: Leveraging segment routing to improve network monitoring," in *Proc. IEEE 35th Annu. Int. Conf. Comput. Commun. (INFOCOM 2016)*, 2016, pp. 1–9.

[30] X. Li and K. L. Yeung, "Bandwidth-efficient network monitoring algorithms based on segment routing," *Comput. Netw.*, vol. 147, pp. 236–245, Dec. 2018.

[31] M. Polverini, A. Cianfrani, M. Listanti, and A. Baiocchi, "Routing perturbation for traffic matrix evaluation in a segment routing network," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1645–1660, Dec. 2018.

[32] A. Cianfrani, M. Polverini, and T. Nalawade, "A heuristic approach to assess the traffic matrix of an ISP exploiting segment routing flexibility," in *Proc. 30th Int. Teletraffic Congr. (ITC 30)*, vol. 1, Sep. 2018, pp. 194–199.

Marco Polverini received the master's degree in telecommunications engineering and the Ph.D. degree in information and communication engineering from the University of Rome La Sapienza in 2010 and 2014, respectively, where he is currently a Research Fellow with the Department of Information, Electronic and Telecommunications Engineering. His main research interests are routing protocols for energy saving in IP networks, network traffic monitoring, and measurement in next generation routing technologies.

Antonio Cianfrani (Member, IEEE) received the master's degree in telecommunications engineering and the Ph.D. degree in information and communication engineering from the University of Rome Sapienza in 2004 and 2008, respectively, where he is currently an Assistant Professor with the DIET Department. His fields of interests include routing algorithms, network protocols, and performance evaluation of software routers and green networks. His current research interests are focused on segment routing and traffic matrix computation. He serves on the Editorial Boards of the IEEE TRANSACTION ON GREEN COMMUNICATIONS AND NETWORKING.

Marco Listanti (Member, IEEE) received the Dr.Eng. degree in electronics engineering from the University of Rome Sapienza in 1980. In 1981, he joined Fondazione Ugo Bordoni, where he was the Leader of the group TLC Network Architecture until 1991. In 1991 he joined the INFOCOM Department, University of Roma La Sapienza, where he is a Full Professor of switching systems. He participated in several international research projects supported by EEC and ESA. He has authored several papers published on the most important technical journals and conferences in the area of telecommunication networks. His current research interests focus on traffic control in IP networks and on the evolution of techniques for optical networking.