Dynamic priority assignment for SLA compliance in service function chains

Frank Wetzels
CWI
Amsterdam, The Netherlands
f.p.m.wetzels@cwi.nl

Hans van den Berg
TNO
The Hague, The Netherlands
j.l.vandenberg@tno.nl

Joost Bosman
TNO
The Hague, The Netherlands
joost.bosman@tno.nl

Rob van der Mei
CWI
Amsterdam, The Netherlands
R.D.van.der.Mei@cwi.nl

***Abstract*—In service function chaining, data flows from a particular application or user travel along a pre-defined sequence of network functions. Appropriate service function chaining resource allocation is required to comply with the service level required by the application. In this paper, we introduce a dynamic priority assignment for flows that compete for service using a particular network function in a chain. Using the recent results of the performance metrics of transient birth–death processes, we analyse this priority assignment and develop an optimal strategy for selecting a (cheap) low- or (expensive) high-priority service, given the flow's service level agreement requirements. A decision table can, thus, be created to facilitate the fast, online priority scheduling of newly arriving flows requesting service.**

***Index Terms*—Service function chain resource allocation, software defined networking, SLA violation duration, priority allocation.**

I. Introduction

In software defined networking (SDN) [1], a de-coupling between the control and data plane is defined at the switching devices in the network. The control plane is moved to a central controller, and the switching nodes are centrally managed [2], [3] through a controller using south-bound protocols [4].

Virtualization is based on the decoupling of a (network) service function (SF) from the underlying hardware, and the corresponding functions are, thus, called virtualized network functions (VNFs). In a network functions virtualization infrastructure (NFVI), the orchestrator deploys and operates VNFs [5]. The NFVI and SDN are complementary as they offer central control of network functions and network functionality, respectively. The flexibility in creating, deleting, and moving VNFs in a network can be supported by overlays and the flexibility in changing the routing and forwarding of the traffic provided by SDN. NFVI provides new possibilities in the network such as the deployment of on-demand (copies of) network functions and the movement of network functions to other locations in the network [6]. By steering traffic to SFs in a specific sequence, a service function chaining (SFC) [7] is created. Several steering methods exist to direct traffic to SFs [8]. For example, a network service header [9] or by pushing the proper flow-labels through south-bound protocols to the switches [10]. An example of an SFC is presented in fig. 1. The SFs applied to a flow are encryption, screening for viruses and malware, and network address translation [11].

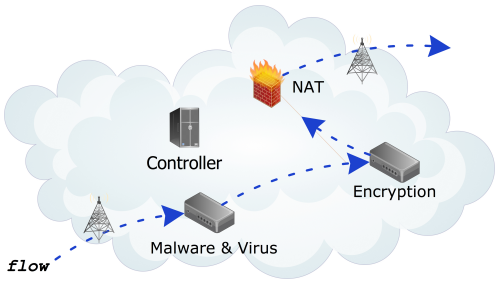


Fig. 1: A service function chain performing malware & virus scanning, encryption and network address translation.

The implementation of SFCs results in new challenges such as chain composition, chain embedding, and scheduling, i.e. service function chaining resource allocation (SFC-RA) or NFV-RA in Internet Engineering Task Force (IETF) and European Telecommunications Standards Institute (ETSI) terminology, respectively. In this paper, we use the IETF terminology.

Many solutions and algorithms have been presented for solving the problems identified in the individual stages of SFC-RA [12]–[14] given a set of constraints, metrics, and service requests (SRs). In reality, SRs are not known in advance and remain in the network for an arbitrary amount of time. In addition, other traffic may be flowing through the SFC. If this traffic varies, it may result in load variations in SFs. We refer to this type of traffic as background traffic. Traffic—that is required to flow through the SFC—due to SRs is considered as high-priority traffic.

Resource allocation (RA) algorithms should be capable of handling SRs upon arrival and of coping with the varying traffic intensity to obtain a realistic RA. This has been called dynamic RA [13]. To obtain a realistic and dynamic RA, the following aspects should be considered: (1) the arrival rate and duration of the SRs, i.e. high-priority traffic, (2) the arrival rate and duration of the background traffic, and (3) the service level agreement (SLA) requirements applicable to high-priority traffic must be met.

In this paper, we focus on dynamic scheduling, where the expected load on an NF is used as a parameter for deciding

whether an NF can handle a newly arrived SR given an SLA. If the SLA requirements can be met, as the expected load on the NF is acceptable, the high-priority traffic is given the same priority as the background traffic. If the SLA requirements cannot be met, the high-priority traffic is directed to an NF that is able to handle high-priority traffic accordingly.

The challenge is to 'predict' whether the load on an NF will result in an SLA violation. To decide how to schedule high-priority traffic, the expected duration of background traffic exceeding a critical level on an NF is determined. If the background traffic level is above the critical level for too long, is expected that the SLA will be violated if high priority traffic is assigned to this NF.

In this paper, a decision rule is presented for assigning a high-priority flow to a normal- or high-priority NF given the SLA. By applying recent mathematical results of performance metrics for transient birth–death (BD) processes [15] to one node, we gain insight regarding the expected duration of a resource violation.

The remainder of this paper is organised as follows. We start with the background and motivation of this paper in section II. In section III, we define our model, the analysis of which is presented in section IV using the recent mathematical results of transient BD processes, and we apply these to our model. The numerical results are presented in section V. We conclude this paper by discussing our findings and propose future works in sections VI and VII, respectively.

II. Background and Motivation

The SFC-RA problem consists of three stages: (i) chain composition, i.e., constructing the sequence of NFs through which flows travel as a result of the SRs, (ii) chain embedding, i.e., the actual deployment of the VNFs in the physical network, and (iii) chain scheduling of the SRs, i.e., proper assignment of flows to VNFs, which are possibly part of multiple chains, while not exceeding the resource constraint(s).

This paper is focused on dynamic online scheduling based on an expected load violation at one NF. The chain composition and chain imbedding phases are considered to be completed at this point. This is not a restriction, given the numerous proposed solutions for these phases in the literature.

An SR, upon arrival, will be scheduled to the NF as long as the expected load violation does not invalidate the SLA. Traffic due to an SR is considered as a high-priority flow in this paper. The other traffic flowing through the network may flow through the NF as well, thereby affecting the load. We refer to this traffic as the background flows. If the SLA is expected to be violated, a high-priority flow is directed to a second NF that is capable of handling the high-priority flow.

To decide how to schedule an SR upon arrival, two BD-processes [16] are defined. One process generates the background flows, while the second process generates a high-priority flow. By considering the expected amount of time a BD process spends above a certain (critical) level [15], the expected duration of the load violation can be determined for an exponentially distributed time interval, i.e. the lifetime of the high-priority flow.

To the best of our knowledge, the expected load on an NF (or NFs) has not been taken into consideration yet.

Throughout this paper, the load on an NF is considered, which represents the delay an NF imposes on individual packets that travel through the NF, i.e. a varying load leads to a varying waiting time for packets to be processed.

In this paper, we present a decision rule for determining whether a newly arrived high-priority flow should be handled in normal or high priority, based on the expected duration of the load violation on an NF during the life-time of the high-priority flow.

III. Model of a single-node SFC

In fig. 2, an SFC is presented. The network consists of the nodes A and B, SF F, and controller C orchestrating the network. A high-priority flow f arrives at node A, passes through F while undergoing a function F_j with priority $j = 1$ or 2, and leaves the network at B. At F, background traffic exists, which is indicated in the figure by the dashed curved arrow that travels through the node undergoing SF F_1 and disappears. While f travels through F, a maximum load should not be violated for too long.

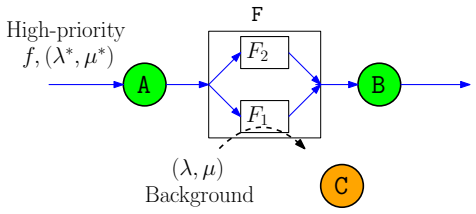


Fig. 2: Network with service node F with background traffic, in which function F_j is applied to flows with priority $j = 1, 2$.

Assumptions:

1) The communication between C and F is not considered.
2) The transmission speed and transmission delay are not considered.
3) C has all the information required to make decisions. C decides if a high-priority flow will be handled by a high-priority function.

Definitions:

1) The background flows arriving at F are driven by a BD process. All the background flows are handled as normal priority flows at F.
2) High-priority flows are driven by the BD process. No more than one high-priority flow is present.
3) Functions F_j can run at priority $j = 1$ (normal) or 2 (high) and not necessarily on one node. F_j has a fixed capacity in serving requests and is not shared with other processes, i.e. F_j are independent.
4) No priority-scheduling exists at F.
5) The following service level applies to high-priority flows: The fraction of time of the load violation that a high-priority flow f is allowed to undergo while traveling

through F is less than α, where α is measured as a fraction of time of the life-time of f for which the maximum load is exceeded. For example, if the load-SLA is 95%, $\alpha = 0.05$ during the life-time of f.

The purpose is to decide upfront, upon arrival of a high-priority flow, at what priority this flow should be processed while meeting the SLA requirements. The decision is based on the expected duration of the SLA violation, given a number of background flows.

IV. Analysis

In this section, a mathematical preparation is presented which enables us to determine the expected duration of the load violation at a node. We will use the results presented in [15]. Based on these results, a maximum number of background flows can be determined for which the load violation remains acceptable.

Two BD processes χ^* and χ are defined, which drive the high-priority flow and the number of background flows, respectively. In this section, we start with an infinitely large state space for χ. This allows us to use the results in [15]. To calculate the solutions, a finite system of linear equations should be solved. Hence, only states $\{0, 1, \ldots, N\}$ are considered. This aligns with a finite number of flows present in the network owing to physical boundaries It should be noted that the state space of χ is not truncated, as that would imply that χ cannot jump to states beyond the state space boundary.

A. Mathematical preparation

Let χ:=$\{X(t) \in S | t \geq 0\}$ and χ^*:=$\{X^*(t) \in S^* | t \geq 0\}$ be BD processes with BD rates λ, μ and λ^*, μ^* and state spaces $S = \{0, 1, \ldots\}$ and $S^* = \{0, 1\}$ which drive the background traffic and high-priority flow f, respectively. It should be recalled that for a BD process χ with BD rates of $\lambda_j = \lambda$ and $\mu_j = j\mu$, the following holds for $j = 0, 1, \ldots$ and $\rho := \frac{\lambda}{\mu}$.

$$\pi_j = e^{-\rho} \frac{\rho^j}{j!}, \tag{1}$$

with π_j refers to the steady state probabilities of χ. The processes χ and χ^* start at the moment the network starts running. If χ^* jumps to 1, f arrives. At that moment, we reset the time for χ and set $X(0) = n$, which is the number of background flows travelling through F. If χ^* jumps to 0, f ends. Let T be the lifetime of f, which is exponentially distributed with mean $\frac{1}{\mu^*}$.

Define the amount of time U_m at which χ, $t \in [0, T]$, spends above some level $m \in S$ during the lifetime of f as follows.

$$U_m := \int_0^T \mathbf{1}_{\{X(t) > m\}} \, dt. \tag{2}$$

We are interested in the expected time that χ spends above level $m \in S$ during the lifetime of f, given the number of background flows n at F at the moment f arrives (starts). The conditional expected cumulative residential time is defined as,

$$\mathbb{E}(U_m | X_0 = n) = \sum_{j=0}^{\infty} \mathbb{E}(U_m, X_T = j | X_0 = n), \tag{3}$$

with $X_0 := X(0)$ and $X_T := X(T)$. Define the following Laplace-Stieltjes transfrom (LST) for $n, j, m \in S$,

$$\tilde{K}_{m,n,j}(s) := \mathbb{E}(e^{-sU_m} \mathbf{1}_{\{X_T = j\}} | X_0 = n). \tag{4}$$

$\tilde{K}_{m,n,j}$ is the LST of the amount of time χ spends above m intersected by χ is at state j at $t = T$ conditioned on χ starts in n. In [15], a procedure is presented to determine (4). To determine (4), only the states $\{0, 1, \ldots, N\}$ are considered, thus preventing solving an infinite system of linear equations. We, thus, define $S_N := \{0, 1, \ldots, N\}$. That is, χ may jump at states beyond N. However, these states are not included in the calculations below.

It was found that, for certain $m \in S_N$ the following finite system of $N + 1$ linear equations should to be solved in the Laplace domain.

$$\tilde{K}_m^N \phi_{m,n} = \mu^* e_j, \tag{5}$$

for each $m, n, j \in S_N$. e_j is an $N+1$ dimensional unit-vector, and $\phi_{m,n}$ is an $N + 1$ dimensional vector of which the j-th component, $(\phi_{m,n})_j$, is the LST of the time χ, which resides above m, while $X_T = j$ given that $X_0 = n$. $\tilde{K}_m^N$ is represented by a $(N + 1) \times (N + 1)$ matrix κ_m with,

$$(\kappa_m)_{n,k} = \begin{cases} \lambda_n + \mu_n + \mu^* + s\mathbf{1}_{\{n>m\}}, & k = n, \\ -\lambda_n \mathbf{1}_{\{n \leq m\}}, & k = n + 1, \\ -\mu_n, & k = n - 1, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

Define $E_{m,n} := \mathbb{E}(U_m | X_0 = n)$, as given by (3). We are interested in $E_{m,n}$ restricted to S_N. Then, (5) should be solved for each $m, n, j \in S_N$. On applying Cramer's rule we can determine vector $\phi_{m,n}$ for each $m, n \in S_N$ (the inverse of κ_m exists [15]). For $m, n \in S_N$, define $E_{m,n}^N$ as the result obtained on implementing the above procedure while calculating $E_{m,n}$. Then,

$$E_{m,n}^N := -\sum_{j=0}^{N} \lim_{s \to 0} \frac{d}{ds} (\phi_{m,n})_j(s). \tag{7}$$

For $m, n \in S_N$, define $T_{m,n}^N$ as the fraction of time during the lifetime of f for which χ is in states above m with $X_0 = n$. Then,

$$T_{m,n}^N := \frac{1}{T} E_{m,n}^N. \tag{8}$$

To calculate $T_{m,n}^N$, we divide the left-hand side of (7) by T, as T can be considered as an instantiation of the duration of f.

B. Priority assignment rule

A maximum load on the SF F corresponds to a maximum number of background flows, say m. If $m + 1$ (or more) flows would arrive at F, the maximum load will be exceeded. However, if the duration of the presence of $m+1$ or more flows during the lifetime of f meets the SLA, the load violation is acceptable. Let $\hat{n}$ be the maximum number of background

flows to have f processed by a normal-priority function during the lifetime of f without violating the SLA. Then,

$$\hat{n} = \max\{n \in S_N | T^N_{m,n} < \alpha\}. \quad (9)$$

By determining $T^N_{m,n}$ for all $m, n = 0, 1, \ldots, N$, $\hat{n}$ can be found by searching for n for which the greatest value of $T^N_{m,n}$ is less than the expected duration of the load violation. As a result, a $(\alpha, \hat{n})$ lookup table can be created which lists $\hat{n}$ given the SLA. When f starts, whether f can be handled by a normal-priority function based on the SLA and number of background flows n is decided. If $n > \hat{n}$ at the moment f starts, f should be handled by a high-priority function. Otherwise, f can be handled by a normal priority function. It should be recalled that the $(\alpha, \hat{n})$ look-up table is based on the characteristics of the high-priority and background traffic, i.e. the BD parameters of the BD processes.

V. Numerical results

The procedure in the previous section is applied to the following situation. The SFC consists of one node, and the number of background flows is restricted to some value N, i.e. we truncate the state space to N. In section V-A, the outcome of the examples is presented, and in section V-B, the results obtained on applying the decision rule are discussed. The truncation effect, presented in the examples below, is discussed first.

It should be recalled that U_m is concerned with the time of χ spent *above* state m. As a result of truncating the state space to N, the above calculations result in zero contributions to $E^N_{m,n}$ for the states above N. The difference between $E_{N-1,n}$ and the calculated value $E^N_{N-1,n}$ increases if n gets 'closer' to N.

A. Performance results

Let us consider the following examples. The high-priority flow f has a duration of $0.01, 10$, and 1000 s. The relative expected conditional residential times $T^N_{m,n}$, where $\rho = 1, 5, 10$ ($\mu = 0.2$ and $\lambda = 0.2, 1.0, 2.0$) are determined by solving (5). We set the size of the truncated state space to 25, i.e. $S_N = \{0, 1, \ldots, 24\}$.

1) High-priority flow duration of 10 s: Refer to fig. 3. The duration of f is set to 10 s. Figs. 3a-3c show $T^{25}_{X_0,m}$ for $\rho = 1, 5$, and 10, respectively. The moment f started X_0 background flows were present. Obviously, for given X_0, $T^{25}_{X_0,m} > T^{25}_{X_0,m+1}$, as χ spends longer in states $m, m+1, \ldots$ than in states $m+1, m+2, \ldots$. For a given m, the contribution to the residential time is zero as long as χ is at states $0, 1, \ldots, m$ and positive as long as χ spends at states $m+1, m+2, \ldots$. The greatest change in $T^{25}_{m,n}$ occurs if $X_0 = m$ and χ jumps to $X_0 + 1$ or $X_0 = m + 1$ and χ transitions to $X_0 - 1$, as χ immediately contributes or stops contributing to $T^{25}_{m,n}$, respectively. Whereas, the larger $|m - X_0|$ is, the longer it takes for χ to jump into the states that contribute (or stop contributing) to U_m.

As ρ increases, χ spends a greater amount of time at higher states. As a result, T^{25}_{m,X_0} increases.

2) High-priority flow duration of 1000 s: Refer to fig. 4. The duration of f is 1000 s. Figs. 4a–4c show T^{25}_{m,X_0} for $m = 0, 1, \ldots, N$ and $\rho = 1, 5$, and 10, respectively. Process χ reaches its steady state as T^{25}_{m,X_0} is independent of X_0, while disregarding the truncation effect. For example, fig. 4a shows that $T^{25}_{0,X_0} \approx 0.63$, i.e. the procedure in section IV-A determines that during 63% of the lifetime of f, χ is in states $\{1, 2, \ldots, N\}$.

Define R_m as the relative residential time of χ in states $\{m+1, m+2, \ldots, N\}$ while χ is in steady state, i.e. $T \to \infty$. Then,

$$R_m := \sum_{j=m+1}^{N} \pi_j, \quad (10)$$

with π_j given by (1). As per [15], appendix D, $\frac{U_m}{T}$ converges almost surely to R_m. In table I, R_0, R_1 and R_2 are given for $\rho = 1, 5, 10$.

m	$\rho = 1$	$\rho = 5$	$\rho = 10$
0	$R_m \approx 0.632$	$R_m \approx 0.993$	$R_m \approx 0.999$
1	$R_m \approx 0.264$	$R_m \approx 0.960$	$R_m \approx 0.999$
2	$R_m \approx 0.080$	$R_m \approx 0.875$	$R_m \approx 0.997$

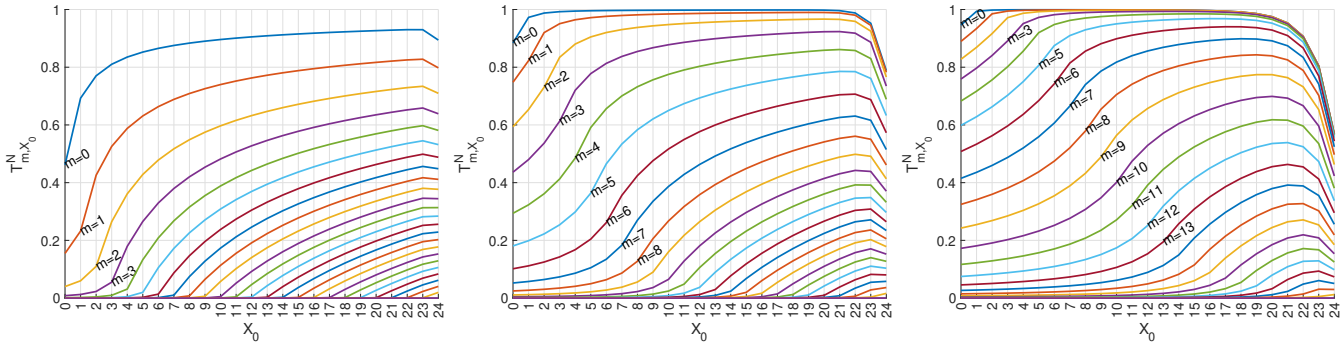
TABLE I: Relative residential time of χ in states $\{m+1, m+2, \ldots, N\}$ while in steady-state for $\rho = 1, 5, 10$ and $m = 0, 1, 2$.

The values for R_0, R_1, and R_2 are consistent with $T^{25}_{0,X_0}, T^{25}_{1,X_0}$, and T^{25}_{2,X_0}, respectively, in figs. 4a and 4b for $\rho = 1, 5$. However, in fig. 4c, we observe slightly lower values for $T^{27}_{0,X_0}, T^{27}_{1,X_0}$, and T^{27}_{2,X_0} than the expected values (table I). The diagram for $N = 25$ (not shown) showed lower values for $T^{25}_{0,X_0}, T^{25}_{1,X_0}$, and T^{25}_{2,X_0} for $\rho = 10$. Therefore, in fig. 4c the results for $N = 27$ are shown. This illustrates that enlarging the truncated state space, 'postpones' the truncation effect.

3) High-priority flow duration of 0.01 s: The duration of f is 0.01 s. In fig. 5, T^{25}_{m,X_0} is shown for $\rho = 5, \lambda = 1.0$. As the duration of f is very short, the 'movement' of the BD process χ is limited to values 'around' its starting value X_0. Set $m = X_0$, i.e. the level of interest is equal to the number of background flows. Then, for $n \leq m$, we have $T^{25}_{m,n} \approx 0$, as during this short period of time, χ is in states $\{0, 1, \ldots, m\}$ for the majority of the time. For $n > m$, we have $T^{25}_{m,n} \approx 1$, as during this short period of time, χ is in states $\{m+1, m+2, \ldots, N\}$ for the majority of the time. This also holds for $\rho = 1, 10$ (not shown). Hence, the results for $\rho = 1$ and $\rho = 10$ appear to be similar to fig. 5.

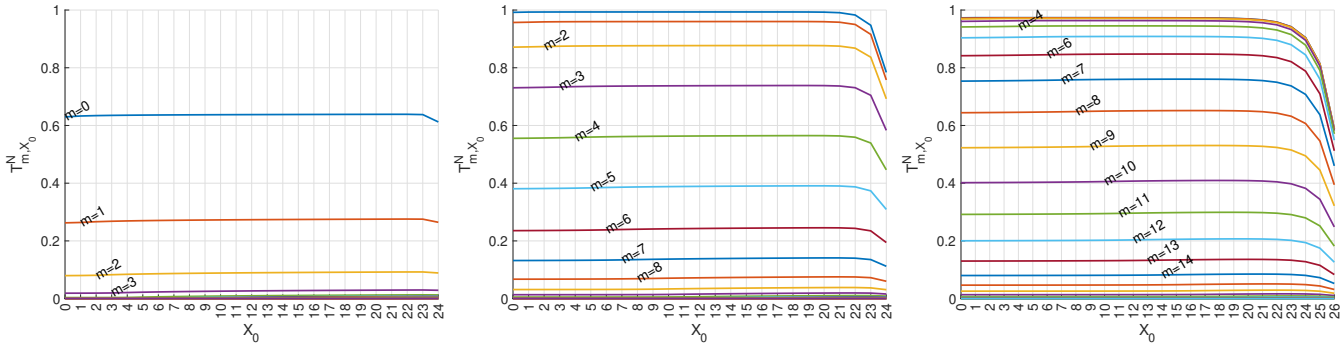
B. Priority assignment results

In section V-A1, we determined T^{25}_{m,X_0} for $X_0, m \in S_N, \rho = 5, 10$ and $T = 10$ s. Based on these results, we can determine $\hat{n}$ by determining the largest value



(a) $N = 25, \rho = 1, \lambda = 0.2$ (b) $N = 25, \rho = 5, \lambda = 1.0$ (c) $N = 25, \rho = 10, \lambda = 2.0$

Fig. 3: Relative conditional expected cumulative residential time $T^{25}_{X_0,m}$ for $T = 10$ s.



(a) $N = 25, \rho = 1, \lambda = 0.2$ (b) $N = 25, \rho = 5, \lambda = 1.0$ (c) $N = 27, \rho = 10, \lambda = 2.0$

Fig. 4: Relative conditional expected cumulative residential time $T^{25}_{X_0,m}$ and $T^{27}_{X_0,m}$ for $T = 1000$ s.

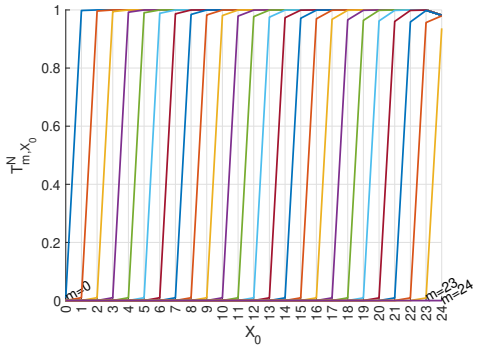


Fig. 5: Relative conditional expected cumulative residential time T^{25}_{m,X_0} for $T = 0.01$ s and $\rho = 5, \lambda = 1.0$.

of n for which $T^N_{m,n} < \alpha$ holds for given m. In fig. 6, $\hat{n}$ versus α for $\rho = 10$ and a maximum load m is given. We select $m = 13, 15$, and 20. For example, in fig. 6a for $\alpha = 0.05$, the number of acceptable background flows is 12 (with a maximum load of 15), when f starts. This means that if $n \leq 12$, f should be considered as normal priority. Otherwise it should be considered as high-priority.

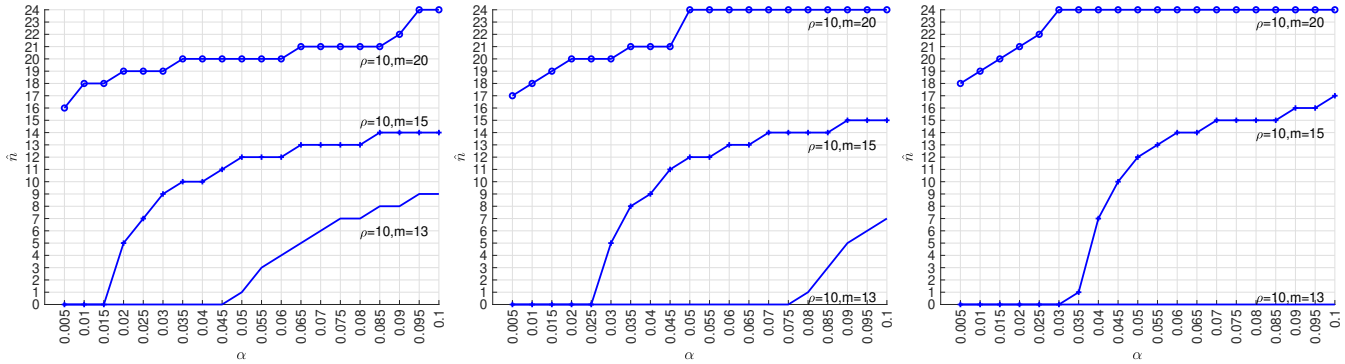
1) High-priority flow duration of 10 and 20 s: Figs. 6 and 7 show $\hat{n}$ for $\rho = 10$ with different BD parameters for $T = 10$ s and $T = 20$ s, respectively. The figures show an increased traffic dynamics of the background flows, i.e. the flows set up and disappear faster. As T increases, χ can jump at states above m longer. Therefore, if the same SLA is applied to different values of T, a smaller number of background flows is acceptable as T increases.

2) High-priority flow duration of 1000 s: In fig. 8, the $\hat{n}$ is shown for $T = 1000$ s. The $\hat{n}$-table presents the steady state of χ. Using (8), we first obtain $T^{27}_{20,X_0} < 0.005$ (strongest SLA), i.e. regardless of the number background flows, the given SLAs will be met. Second, $T^{27}_{13,X_0} > 0.1$ (weakest SLA), i.e. no given SLA can be met. Third, $0.05 > T^{27}_{15,X_0} > 0.045$. Hence, there exists a the steep ramp at $\alpha = 0.045$.

3) High-priority flow duration of 1 s: In fig. 9, $\hat{n}$ is given for $T = 1$ s, $\rho = 10$ and $\mu = 0.2$. In such a case, the T^{25}_{m,X_0} diagram comprises a 'S' shaped graph (not shown) for all m. Owing to of the steep graphs for T^{25}_{m,X_0}, $\hat{n}$ will not vary much. Hence, $\hat{n}$ increases slowly in fig. 9.

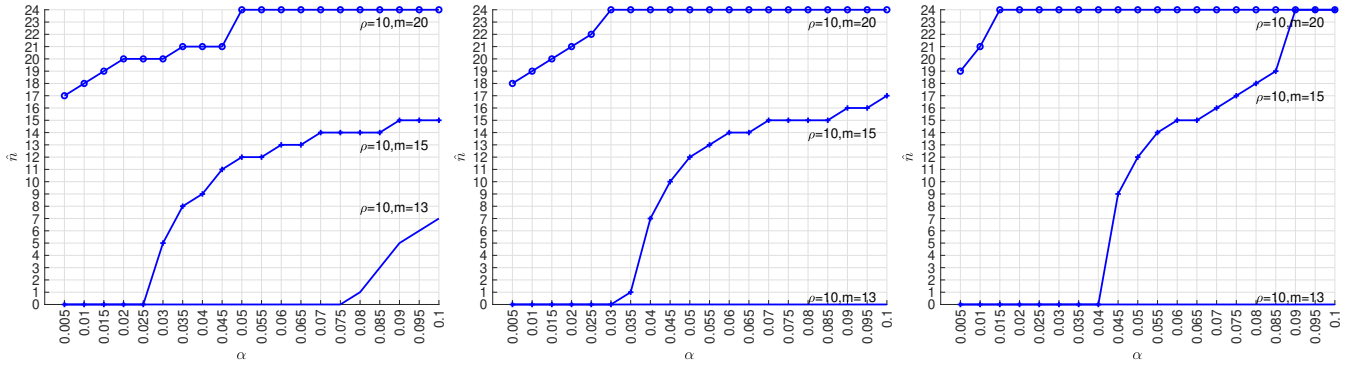
4) General remarks on priority assignment: In general, if the SLA deteriorates, $\hat{n}$ increases, as the duration for which χ is allowed to stay at states greater than m increases. Secondly, with an increasing maximum load m, $\hat{n}$ increases. This can be observed in all the figures.

An increasing value of λ and μ while ρ remains constant



(a) $T = 10, \rho = 10, \mu = 0.2$ (b) $T = 10, \rho = 10, \mu = 0.4$ (c) $T = 10, \rho = 10, \mu = 0.8$

Fig. 6: Maximum number of background flows ($\hat{n}$) versus the SLA applied to a high-priority flow with a duration of $T = 10$ s.



(a) $T = 20, \rho = 10, \mu = 0.2$ (b) $T = 20, \rho = 10, \mu = 0.4$ (c) $T = 20, \rho = 10, \mu = 0.8$

Fig. 7: Maximum number of background flows ($\hat{n}$) versus the SLA applied to a high-priority flow with a duration of $T = 20$ s.

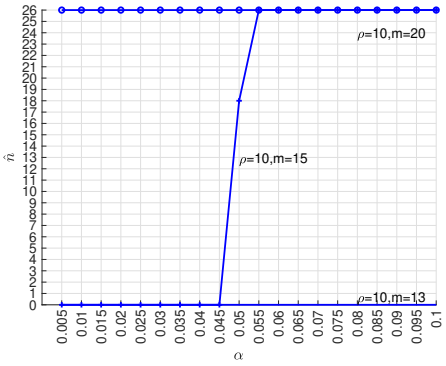


Fig. 8: Maximum number of background flows ($\hat{n}$) versus the SLA applied to a high-priority flow with $\rho = 10, \mu = 0.2$, $T = 1000$ s, and $N = 27$.

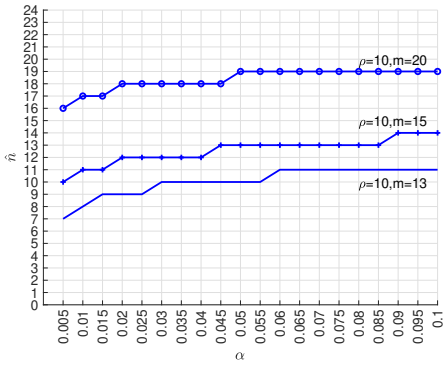


Fig. 9: Maximum number of background flows ($\hat{n}$) versus the SLA applied to a high-priority flow with $\rho = 10, \mu = 0.2$ and $T = 1$ s.

indicates that flows start and disappear faster on average, i.e. the traffic becomes more bursty. The results show a more 'vertical shape' of $\hat{n}$. This might be explained as follows: ρ represents the long-run average number of background flows. Figs. 6 and 7 do not show the long-run situation. However, they suggest that if the traffic becomes more bursty, the variance of the number of background flows increases. However, further research is required on this subject.

VI. Discussion and Conclusion

The computation of the above results may be time consuming for a large state space. Given a state space of size N,

for each $m \in S_N$, a system of N linear equations is solved. We developed the software in MATLAB to determine the expected load violation. As the calculations involve symbolic manipulations while applying Cramer's rule, the calculations required, lasted too long to be applied online in a network on a controller or orchestrator.

A small truncated state space is not an issue when applied to very large data transports such as backups or 3D video streams. The bandwidths used by these streams may be very large, such that the number of concurrent streams is limited despite the huge bandwidths available in modern networks nowadays.

By calculating a $\hat{n}$-table in advance and implementing it as a lookup-table, the decision making becomes very fast and would fit well in an SDN.

We conclude by stating that the determination of the expected duration of the load violation provides an operator with the possibility of deciding upfront if a high-priority flow should be assigned to a high-priority function given an SLA. As a result, an operator is able to use its network more efficiently by selecting alternate paths upfront if a high-priority flow starts.

VII. Future work

In the present paper, the focus was on a single node in a SFC. Obviously, an SFC may consist of more than one node. In such a case, the end-to-end delay requirement over all the nodes should be met. Determining if and at what node(s) a high-priority flow can be processed by a normal-priority function (or should be processed at high priority) is a complex joint optimization problem that should be addressed as a follow up to our current analysis.

When considering voice or video traffic as the high-priority flow, the load variability is an important factor. We have suggested that background traffic burst affects the load variability. It is suggested that further research be conducted on load variability and setting the load variability as an additional requirement.

In section V-A, we observed the effect of the truncation of the state space. The probability of leaving the truncated state space increases as the critical level m and starting state X_0 get closer to the truncated state space boundary. The BD process χ still moves, including at states beyond the truncated state space. With N being the truncated state space boundary, it is suggested state $N+1$ be replaced by $N'+1$, for representing all states $\{N+1, N+2, \ldots\}$. However, the basis of the calculations presented in [15] may no longer be applicable, as we then obtain a BD process with a boundary. The applicability of [15] may have to be expanded to other processes as well.

References

[1] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," *IETF Network Task Force*, no. rfc 7426, January 2015.

[2] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," *IETF Network Task Force*, no. rfc 6241, June 2011. [Online]. Available: http://www.rfc-editor.org/info/rfc6241

[3] B. Pfaff and B. Davie, "The Open vSwitch Database Management Protocol," *IETF Network Task Force*, no. rfc 7047, December 2013.

[4] "OpenFlow Switch Specification," *Open Networking Foundation*, no. ONF TS-025, March 2015, version 1.5.1.

[5] ETSI Industry Specification Group (ISG), "Network Functions Virtualisation (NFV); Infrastructure Overview," 2015.

[6] B. Yi, X. Wang, K. Li, S. k.Das, and M. Huang, "A comprehensive survey of Network Function Virtualization," *Computer Networks*, vol. 133, pp. 212–262, Mar. 2018.

[7] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," *IETF Network Task Force*, no. rfc 7665, 2015.

[8] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic Steering for Service Function Chaning," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 487–507, Aug. 2018.

[9] P. Quinn, U. Elzur, and C. Pignataro, "Network Service Header (NSH)," *IETF Network Task Force*, no. rfc 8300, January 2018.

[10] F. Callegati and W. Cerroni and C. Contoli and G. Santandrea, "Dynamic chaining of Virtual Network Functions in cloud-based edge networks," *Proceedings of the 2015 1st IEEE Conference on Network Softwarization*, June 2015.

[11] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," *IETF Network Task Force*, vol. rfc 3022, January 2001.

[12] Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service Function Chaining Resource Allocation: A Survey," *ArXiv e-prints*, Jul. 2016.

[13] J. G. Herrera and J. F. Botero, "Resource Allocation in NFV: A Comprehensive Survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, September 2016.

[14] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A Survey on Service Function Chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138–155, Nov. 2016.

[15] W. Ellens, M. Mandjes, J. van den Berg, D. Worm, and S. Blaszczuk, "Performance valuation using periodic system-state measurements," *Performance Evaluation*, vol. 93, pp. 27–46, 11 2015, eemcs-eprint-26919.

[16] L. Kleinrock, *Queueing Systems, Volume I: Theory*. Wiley & Sons, 1975.