

A Lightweight, Non-intrusive Approach for Orchestrating Autonomously-managed Network Elements

Christos Liaskos, *Member, IEEE*

Abstract—Software-Defined Networking enables the centralized orchestration of data traffic within a network. However, proposed solutions require a high degree of architectural penetration. The present study targets the orchestration of network elements that do not wish to yield much of their internal operations to an external controller. Backpressure routing principles are used for deriving flow routing rules that optimally stabilize a network, while maximizing its throughput. The elements can then accept in full, partially or reject the proposed routing rule-set. The proposed scheme requires minimal, relatively infrequent interaction with a controller, limiting its imposed workload, promoting scalability. The proposed scheme exhibits attracting network performance gains, as demonstrated by extensive simulations and proven via mathematical analysis.

Index Terms—software-defined networking, traffic engineering, backpressure routing.

I. INTRODUCTION

SOFTWARE-Defined Networking can imbue the network management process with an unparalleled level of state monitoring and control. The ability to migrate the routing elements of a network from closed, static hardware solutions towards an open, re-programmable paradigm is expected to promote significantly the adaptivity to demand patterns, eventually yielding a healthy and constant innovation rate. The OpenFlow protocol and assorted hardware [1], which enables an administrative authority to centrally monitor a network and deploy fitting routing strategies, has produced significant gains in a wide set of application scenarios [2], [3].

Nonetheless, SDN-enabled traffic engineering (TE) approaches are presently characterized by a high degree of architectural penetration. Each networking element must yield its inner operation to a remote, central controller. While this assumption is valid for networks managed by the same authority (e.g. [2], [4]), it poses an issue for networks comprising self-managed elements. Furthermore, related solutions may come at a high capital cost, requiring multiple powerful controllers to cover a network [5], as well as a high operational cost, incurred by the need for close interaction between the network elements and the controller [6], which naturally translates to traffic overheads. These concerns, combined with point-of-failure and security considerations [7], can discourage self-managed elements for adopting or even trying an SDN-based, central traffic orchestration.

The present study claims that a lightweight TE solution is in need in order to demonstrate the gains of SDN-enabled collaboration and gradually convince self-managed elements to participate further. The methodology consists of applying the principles of

Backpressure routing [8] to a backbone network of self-managed nodes, deriving stability-optimal flow routing rules. Nodes that choose to participate to the proposed scheme initially inform a central controller of their aggregate, internal congestion states. In return, they receive the aforementioned rule set in the form of a proposal. Apart from its simplicity and ability to respect peering agreements, the proposed scheme also fills a theoretical gap in the related work, offering *analytically*-proven throughput optimality and network stabilization potential.

II. RELATED WORK

Studies on traffic engineering in networks, whether SDN-enabled or not, target the real-time grooming of data flows, in order to provide the best possible quality of service on a given physical infrastructure. To this end, maximizing the network's throughput has constituted a prominent goal. MicroTE [9], Hedera [10] and Mahout [11] focus on the detection and special handling of large "elephant" flows, under the assumption that they constitute the usual suspects of congestion. When a large flow is detected, it is treated as a special case, and it is assigned a separate path, which does not conflict with the bulk of the remaining traffic. These schemes require constant monitoring of the network's state, which is achieved by scanning the network for large flows via periodic polling (at the scale of $5sec$), raising SDN controller scalability and traffic overhead concerns. They differ, however, in where the scanning takes place. Hedera constantly scans the edge switches of the network, requiring less nodes to visit but more flows per node to scan. Mahout scans the hosts, scanning on average more nodes than Hedera, but with less flows to be monitored per node. Finally, MicroTE relies on push-based network monitoring, with nodes posting periodically their state to the controller.

Companies have also invested in SDN-powered solutions for optimizing their proprietary networks, within or among data-centers. Emphasis is placed on prioritizing the applications and flows that compete for bandwidth, based on their significance or operational requirements. B4 [4] incorporates this concern by keeping tuples of source, destination and QoS traits per network flow. The network's resources are constantly monitored and the flows are assigned paths according to their priority, breaking ties in a round-robin manner. Microsoft's SWAN [2] considers classes of priorities, pertaining to critical-interactive, elastic and background traffic. Resources are first assigned per priority class. Within each coarse assignment, a max-min fairness approach is used to distribute resources to specific flows. Bell Labs propose a more direct approach, seeking to solve the formal link utilization problem, given explicit flow requests [3]. Other studies focus on scenarios such as partially SDN-controlled networks, or advancing the efficiency of multipath routing beyond classic approaches [12], exploiting the monitoring capabilities of OpenFlow [13], [14].

C. Liaskos is with the Aristotle University of Thessaloniki, Department of Computer Science, GR-54124, AUTH Campus, Greece and with the Foundation for Research and Technology Hellas (FORTH), Institute of Computer Science, N. Plastira 100, Vassilika Vouton, GR-70013 Heraklion, Crete, Greece. e-mails: cliaskos@csd.auth.gr, cliaskos@ics.forth.gr.

Differentiating from the outlined studies, the present work proposes a SDN-enabled traffic engineering approach that is considerably more lightweight in terms of overhead, as well as less intrusive in terms of architecture. Its goal is to encourage centralized, SDN-based orchestration among autonomously managed networked elements. The proposed scheme is throughput-optimal, yields minimal interaction with the controller and minimal number of required flow rules.

III. PREREQUISITES AND SYSTEM MODEL

An important term in networking studies is the notion of *network stability*. It is defined as the ability of a routing policy to keep all network queues bounded, provided that the input load is within the network's traffic dispatch ability, i.e. within its *stability region*. With $U_{(n,c)}(t)$ denoting the aggregate traffic accumulated within a network node n at time t , destined towards node c , stability is formally defined as [15, p. 24]:

$$\limsup_{\tau \rightarrow +\infty} \frac{1}{\tau} \sum_{t=1}^{\tau} E \{U_{(n,c)}(t)\} < \infty, \forall n, c \quad (1)$$

where τ is the time horizon and $E \{*\}$ denotes averaging over any probabilistic factors present in the system.

A well-developed framework for deducing network stability under a given network management policy is the Lyapunov Drift approach. It defines a quadratic function of the form:

$$L(t) = \sum_{\forall n} \sum_{\forall c} U_{(n,c)}^2(t) \quad (2)$$

The goal is then to deduce the bounds of $\Delta L(t) = E \{L(t+T) - L(t)\}$, which describes the evolution of the network queue levels over a period T . The *Lyapunov stability theorem* states that if it holds:

$$\Delta L(t) \leq B - \epsilon \cdot \sum_{\forall n} \sum_{\forall c} U_{(n,c)}(t) \quad (3)$$

for two positive B, ϵ quantities, then the network is stable and average queue size of inequality (1) is bounded by B/ϵ instead of drifting towards infinity.

The *backpressure algorithm* (BPR) defines a joint scheduling-routing algorithm that complies with the stability criteria of inequality (3) and, most importantly, has been proven to be throughput optimal [16]. Its goal is to minimize the lower bound of $\Delta L(t), \forall t$, effectively suppressing the average queue level within the network. The analytical approach, followed by related studies [17], is based on the queue dynamics expressed by the following relation:

$$U_{(n,c)}^{(t+T)} = \max \left\{ 0, U_{(n,c)}^{(t)} - O_{(n,c)}^{t \rightarrow t+T} \right\} + I_{(n,c)}^{t \rightarrow t+T} + G_{(n,c)}^{t \rightarrow t+T} \quad (4)$$

where $O_{(n,c)}^{t \rightarrow t+T}$, $I_{(n,c)}^{t \rightarrow t+T}$ and $G_{(n,c)}^{t \rightarrow t+T}$ denotes outgoing, incoming and locally generated data at time interval $t \rightarrow t+T$. The usual methodology then dictates a series of relaxations of the right part of eq. (4), based on the following inequalities:

$$O_{(n,c)}^{t \rightarrow t+T} \leq \sum_{l: \text{source}(l)=n} \int_t^{t+T} \mu_l^{(c)}(t) \cdot dt \quad (5)$$

$$I_{(n,c)}^{t \rightarrow t+T} \leq \sum_{l: \text{destination}(l)=n} \int_t^{t+T} \mu_l^{(c)}(t) \cdot dt \quad (6)$$

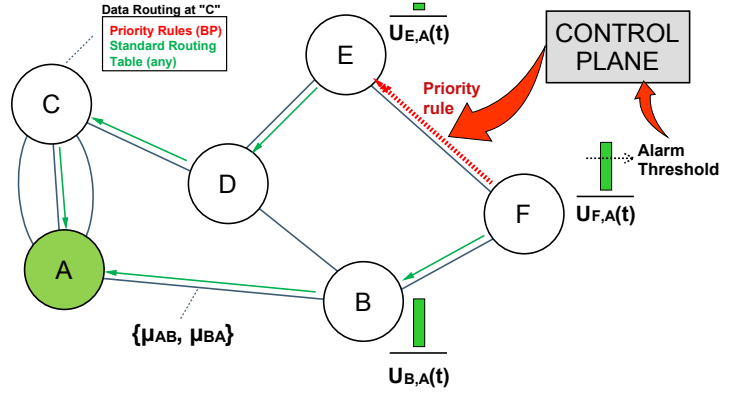


Figure 1. The employed system setup. A network of autonomously managed elements, A-F, uses Backpressure-derived flow rules on top of its standard routing scheme, in order to mitigate congestion events. A centralized control plane orchestrates the operation of the system.

where $\mu_l^{(c)}(t)$ is the maximum allowed bitrate over a network link l carrying traffic destined to node c . Squaring both sides of eq. (4) and incorporating relaxations (5), (6), as well as the identity:

$$V \leq \max \{0, U - \mu\} + A \Rightarrow V^2 \leq U^2 + \mu^2 + A^2 - 2U \cdot (\mu - A) \quad (7)$$

one derives an inequality of the form of relation (3). Further relaxation by substituting all $\mu_l^{(c)}$ and $G_{(n,c)}^{t \rightarrow t+T}$ with maximum allowed values yields compliance with the Lyapunov stability theorem. Furthermore, it is deduced that the upper bound of relation (3) can be minimized when maximizing the quantity:

$$\sum_{\forall n, k, c} \mu_{l: \text{source}(n) \rightarrow \text{dest}(k)}^{(c)}(t) \cdot (U_{(n,c)}(t) - U_{(k,c)}(t)) \quad (8)$$

The standard backpressure routing process, summarized for reference as the SBPR Algorithm, expresses the optimization pursuit of relation (8). According to SBPR, at timeslot $t \rightarrow t+T$, each network link l must carry data towards node c_l^* , such that:

$$c_l^* \leftarrow \operatorname{argmax}_c \{U_{\text{source}(l)}^{(c)}(t) - U_{\text{dest}(l)}^{(c)}(t)\} \quad (9)$$

Bidirectional links are considered as two separate unidirectional links. Originally meant for use in wireless ad hoc networks, the BPR process and its variants have found extensive use in packet switching hardware and satellite systems due to their throughput optimality trait [17]. SBPR variants have adopted latency considerations as well. Most prominently, authors in [18] restrict the node selection step (9) of SBPR only within a subset of links that offer a bounded maximum number of hops towards the target. Other studies have shown that simply altering the queueing discipline from FIFO to LIFO yields considerable latency gains [19]. Finally, it is worth noting that SBPR can be easily made TCP compatible [20].

A. System Model

The present paper studies the use of BPR-variants in backbone networks. The assumed setup, given in Fig. 1, considers a network comprising autonomously managed elements. A node can represent a single physical router or a complete subnetwork, provided that it supports a self-inspecting mechanism for monitoring its internal congestion levels, as well as support a flow-based routing

scheme. The nodes are connected with links of known, time-invariant bandwidth and can be asymmetric or unidirectional with no restriction. Due to this assumption, the notation $\mu_{(n,c)}(t)$ is simplified to $\mu_{(n,c)}$. Data is classified by the originating network identifier (e.g. A), with no further sub-categorization.

The formed network may have any traffic-invariant traffic policy, such as distance vector or shortest path routing. The BPR approach operates on top of the underlying routing scheme and is enforced by a centralized controller, which can receive node state information and propose the installation of priority flow rules. An example is shown in Fig. 1. At time moment t , the controller has assembled a snapshot of the network's state and notices that $U_{(F,A)}(t)$ at node F exceeds a predefined alarm threshold. A BPR-variant is executed, which deduces that traffic from F towards A should better be offloaded to neighboring node E for the time being. A corresponding routing instruction is given to node F , which takes precedence over all other routing rules pertaining to link l_{FE} . Operation is then resumed until the next network state snapshot is received.

OpenFlow-based solutions are most prominent candidates for the control plane and the interaction with the network nodes [1]. In this case, network monitoring can be accomplished by several polling techniques [21], [22]. Without loss of generality, we will assume that the controller obtains a consistent network state with period T [23].

Peering agreements and routing preferences among nodes are also allowed. For example, returning to the example of Fig. 1, the controller would not propose the illustrated flow rule if it was disallowed by the peering policy/agreement between F and E . In other words, when the BPR-variant searches for neighbors $s \in S : \{U_{(s,A)}(t) < U_{(F,A)}(t)\}$, the search is assumed to be limited to nodes that comply to any form of policy, preference of agreement.

Finally, targeting minimal controller load, we allow for at most one priority flow rule per physical network link.

IV. ANALYSIS

We begin the analysis by simplifying the RHS of relation (6), based on the fact that the network links have time-invariant bandwidth:

$$I_{(n,c)}^{t \rightarrow t+T} \leq \sum_{l: d(l)=n} \int_t^{t+T} \mu_l^{(c)} dt = T \cdot \sum_{l: d(l)=n} \mu_l^{(c)} \quad (10)$$

The RHS of relation (5) is simplified even further, given that all traffic from a node n towards a given destination c is served by a single outgoing link, regardless of the enforcement of any BPR priority rules:

$$O_{(n,c)}^{t \rightarrow t+T} \leq \sum_{l: s(l)=n} \int_t^{t+T} \mu_l^{(c)} dt = T \cdot \mu_{l_{nb(n)}}^{(c)} \quad (11)$$

where $b(n)$ is a neighboring node of n complying with any bilateral agreements. Furthermore, applying identity (7) to eq. (4) produces:

$$U_{(n,c)}^2(t+T) \leq U_{(n,c)}^2(t) + \left[O_{(n,c)}^{t \rightarrow t+T} \right]^2 + \left[I_{(n,c)}^{t \rightarrow t+T} + G_{(n,c)}^{t \rightarrow t+T} \right]^2 - 2 \cdot U_{(n,c)}(t) \cdot \left[O_{(n,c)}^{t \rightarrow t+T} - I_{(n,c)}^{t \rightarrow t+T} - G_{(n,c)}^{t \rightarrow t+T} \right] \quad (12)$$

Using the updated relaxations (10) and (11) and setting $\Delta U_{(n,c)}^2(t) = U_{(n,c)}^2(t+T) - U_{(n,c)}^2(t)$ for brevity:

$$\Delta U_{(n,c)}^2(t) \leq T^2 \cdot \left(\mu_{l_{nb(n)}}^{(c)} \right)^2 + \left[T \cdot \sum_{l: d(l)=n} \mu_l^{(c)} + G_{(n,c)}^{t \rightarrow t+T} \right]^2 - 2 \cdot U_{(n,c)}(t) \cdot \left[T \cdot \mu_{l_{nb(n)}}^{(c)} - T \cdot \sum_{l: d(l)=n} \mu_l^{(c)} - G_{(n,c)}^{t \rightarrow t+T} \right] \quad (13)$$

It is not difficult to show that the RHS of inequality (13) can be reorganized as:

$$\Delta U_{(n,c)}^2(t) \leq \left[T \cdot \sum_{l: d(l)=n} \mu_l^{(c)} + U_{(n,c)}(t) + G_{(n,c)}^{t \rightarrow t+T} \right]^2 + \left[T \cdot \mu_{l_{nb(n)}}^{(c)} - U_{(n,c)}(t) \right]^2 - 2 \cdot U_{(n,c)}^2(t) \quad (14)$$

Summing both sides $\forall n, c$ and reminding that $\Delta L(t) = \sum_{\forall n} \sum_{\forall c} \Delta U_{(n,c)}^2(t)$:

$$\Delta L(t) \leq \sum_{\forall n} \sum_{\forall c} \left[T \cdot \sum_{l: d(l)=n} \mu_l^{(c)} + U_{(n,c)}(t) + G_{(n,c)}^{t \rightarrow t+T} \right]^2 + \sum_{\forall n} \sum_{\forall c} \left[T \cdot \mu_{l_{nb(n)}}^{(c)} - U_{(n,c)}(t) \right]^2 - 2 \cdot \sum_{\forall n} \sum_{\forall c} U_{(n,c)}^2(t) \quad (15)$$

We proceed by considering the RHS of relation (15) as a function of the BPR-derived routing decisions $\mu_{l_{nb(n)}}^{(c)}$ and attempt a straightforward optimization. The $\mu_{l_{nb(n)}}^{(c)}$ can be initially treated as continuous variables. Once optimal values have been derived, they can be mapped to the closest of the actually available options within the network topology. The sufficient conditions for the presence of a minimum are:

$$\frac{\partial RHS_{(14)}}{\partial \mu_{l_{nb(n)}}^{(c)}} = 0 \quad (a), \quad \mathcal{H} \left(\frac{\partial RHS_{(14)}}{\partial \mu_{l_{nb(n)}}^{(c)} \cdot \partial \mu_{l_{kb(k)}}^{(c)}} \right) \in \mathbb{R}^+ \quad (b) \quad (16)$$

where k denotes a node, \mathcal{H} is the Hessian matrix [24] and the $0 < \mathcal{H} < \infty$ refers to each of its elements. From condition (16-a) we obtain:

$$T \cdot \left[\sum_{l: d(l)=b(n)} \mu_l^{(c)} + \mu_{l_{nb(n)}}^{(c)} \right] - \left[U_{(n,c)}(t) - \left(U_{(b(n),c)}(t) + G_{(b(n),c)}^{t \rightarrow t+T} \right) \right] = 0, \forall n, c \quad (17)$$

For condition (16-b), it is not difficult to show that it is satisfied due to:

$$\frac{\partial RHS_{(15)}}{\partial \mu_{l_{nb(n)}}^{(c)} \cdot \partial \mu_{l_{kb(k)}}^{(c)}} \propto T^2 > 0, \forall n, k \quad (18)$$

Equation (17) represents a generalization over the SBPR Algorithm, which operates by equation (9). At first, the eq. (17) defines a linear system with discrete variables $\mu_{l_{nb(n)}}^{(c)}$ and can be solved as such. However, interesting approximations can be derived, which

also exhibit the dependence of the optimal solution from the network topology and traffic statistics.

Firstly, the term $T \cdot \left[\sum_{l: d(l)=b(n)} \mu_l^{(c)} + \mu_{l_{nb(n)}}^{(c)} \right]$ represents the aggregate, transit traffic served by node $b(n)$, i.e. the neighbor of n that will be the recipient of traffic destined towards c . A node may assume transit duties in the network, either due to its business logic, or due to its central placement in the topology. On the other hand, the term $\left[U_{(n,c)}(t) - \left(U_{(b(n),c)}(t) + G_{(b(n),c)}^{t \rightarrow t+T} \right) \right]$ refers to the role of node $b(n)$ as generator of new traffic. The quantity $G_{(b(n),c)}^{t \rightarrow t+T}$ also introduces dependence from traffic prediction. Indeed, at time t the controller must obtain an approximation of the traffic that will be generated at node $b(n)$ within the interval $[t, t+T]$. In other words, equation (17) introduces a comparison between the transit and content provider aspects of the network nodes, requiring equally balanced roles. This conclusion is summarized in the following Lemma.

Lemma 1. *Network-wide optimization of throughput requires routing decisions that equalize the transit and content provider roles of the nodes.*

Assuming a network of nodes where data transit prevails over content generation per node, it will hold:

$$\sum_{l: d(l)=b(n)} T \left(\mu_l^{(c)} + \mu_{l_{nb(n)}}^{(c)} \right) > U_{(n,c)}(t) - \left(U_{(b(n),c)}(t) + G_{(b(n),c)}^{t \rightarrow t+T} \right) \quad (19)$$

for all n, c . In this case, the best approach for approximately upholding equation (17) is to maximize the quantity

$$\Delta_{(n,c)}(t) = \left[U_{(n,c)}(t) - \left(U_{(b(n),c)}(t) + G_{(b(n),c)}^{t \rightarrow t+T} \right) \right] \quad (20)$$

which depends on the traffic generated locally at node $n(b)$ during $[t, t+T]$. In other words, the throughput-optimizing routing decision at node n , regarding traffic destined to node c are derived as follows:

$$n^* = \underset{b(n)}{\operatorname{argmax}} \left\{ \Delta_{(n,c)}(t) \right\} \quad (21)$$

where n^* is the optimal neighboring node of n to offload data towards c .

We notice that the transit assumption of (19) is also implied by the SBPR Algorithm. Specifically, SBPR implies that $G_{(b(n),c)}^{t \rightarrow t+T}$ is uniform for all nodes in the network, reducing equation (21) to (9). This limitation is alleviated by the proposed, Foresight-enabled Backpressure Routing (Algorithm 1) which targets backbone networks, where the transit assumption of relation (19) is expected to hold.

Line 5 of the proposed Algorithm reflects the outcome of equation (21). Inspired by [18], we note that line 5 only considers possible nodes c towards which the number of hops does not increase over link l . This approach favors latency and disallows routing loops. If an alarm level is defined, the search in line 5 is restricted further within $c : U_n^{(c)} \geq \text{alarm_level}$. The $\text{visited}[\cdot]$ array is also introduced, to make sure that each possible destination is routed via one link at most, at each node. The optimization of line 10 pertains to the treatment of multi-links that may exist in the network. Assume a triple link $M = \{l_1 : \mu_1, l_2 : \mu_2, l_3 : \mu_3\}$ and a corresponding set of $c_l^*(t)$ assignments $A = \{c_{l_1}^*(t), c_{l_2}^*(t), c_{l_3}^*(t)\}$. Line 10 refers to the optimal reordering of the assignments out of all possible $M \times A$

Algorithm 1 The proposed Foresight-enabled Backpressure Routing algorithm.

```

1: procedure FBPR(network_state( $t = mT | m \in \mathbf{N}$ ))
2:   for each node  $n$  do                                ▷ Define priority flows.
3:      $\text{visited}[c] \leftarrow 0, \forall c$ 
4:     for each link  $l : \text{source}(l) = n$  do
5:        $c_l^*(t) \leftarrow \underset{c: \text{visited}[c]}{\operatorname{argmax}} \{ U_n^{(c)} - (U_{d(l)}^{(c)} + G_{(d(l),c)}^{t \rightarrow t+T}) \}$ 
6:        $\text{visited}[c_l^*(t)] \leftarrow 1$ 
7:        $\Delta Q_l^*(t) \leftarrow \max\{0, U_n^{(c_l^*(t))} - U_{d(l)}^{(c_l^*(t))}\}$ 
8:     end for
9:   end for                                ▷ Consider multi-links, if any.
10:   $\mu^*(t) \leftarrow \underset{\mu}{\operatorname{argmax}} \sum_{\forall l} \mu_l \cdot \Delta Q_l^*(t)$ 
11:  for each link  $l : \Delta Q_l^*(t) > 0$  do
12:    Deploy rule  $\{ \text{from} : s(l), \text{to} : c_l^*(t), \text{via} : l \}$ .
13:  end for
14:  return
15: end procedure

```

combinations and for each multi-link of the network, maximizing the expected throughput. Finally, lines 11 – 13 install the FBPR-derived priority rules to the corresponding nodes.

Corollary 2. *FBPR is throughput-optimal.*

We notice that the preceding analysis takes place before the relaxation of equation (8) of the classic analytical procedure. Applying this final relaxation to equation (15) leads to compliance with the Lyapunov stability criterion (relation (3)) to the proof of throughput optimality, as detailed in [16].

V. SIMULATIONS

In this Section, the performance of the proposed schemes is evaluated in various settings, in terms of achieved average throughput, latency and traffic losses. Specifically, the ensuing simulations, implemented on the AnyLogic platform [25], focus on: i) The performance and stability gains arising from the combination of BPR-based and Shortest Path-based (OPSF) policies, ii) The gains of Foresight-enabled BPR over its predecessors.

The simulations assume 25 autonomously managed nodes, arranged in a 5×5 grid. Each node n_{ij} , $i = 1 \dots 5$, $j = 1 \dots 5$ is connected to its four immediate neighbors, $n_{i+1,j}$, $n_{i-1,j}$, $n_{i,j+1}$, $n_{i,j-1}$, where applicable. This type of topology is chosen to ensure a satisfactory degree of path diversity, i.e. a good choice of alternative paths to connect any two given nodes. We note that path diversity is a prerequisite for efficient traffic engineering in general. Each link connecting two nodes is bidirectional with $2GBps$ bandwidth at each direction.

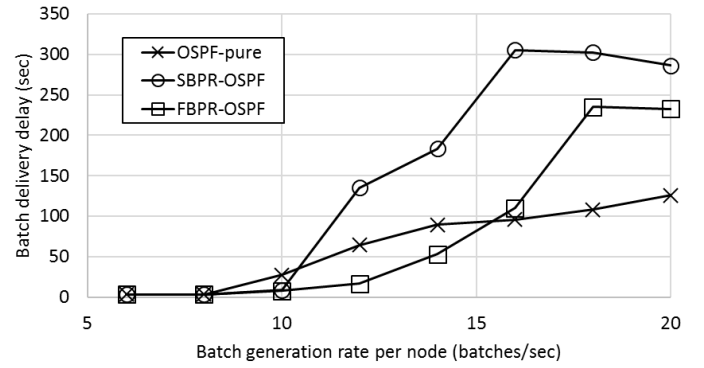
Given that packet-level simulation of backbone networks is not easily tractable in terms of simulation runtimes [10], [11], we assume slotted time (1sec slot duration) and traffic organized in 100MB-long batches. At each slot, a number G_{ij} of batches is generated at each node, expressing concurrent traffic generated from multiple internal users. The destination of each batch is chosen at random (uniform distribution). Then, traffic batches are dispatched according to the routing rules and the channel rates. Each node is assumed to keep track of its internal congestion level and push it with report/actuation period T to a central controller (e.g. like [9]). G_{ij} and T are set or varied per experiment. A

node is assumed to reject/drop incoming or generated traffic when it has more than 500 batches on hold, using a single queue model. Finally, the BPR schemes are enabled on a node where the number of batches on hold exceed a certain *alarm level*, set per experiment. The *alarm_level* can also be perceived as a parameter that defines whether the adoption of the BPR priority flow rules is partial or global.

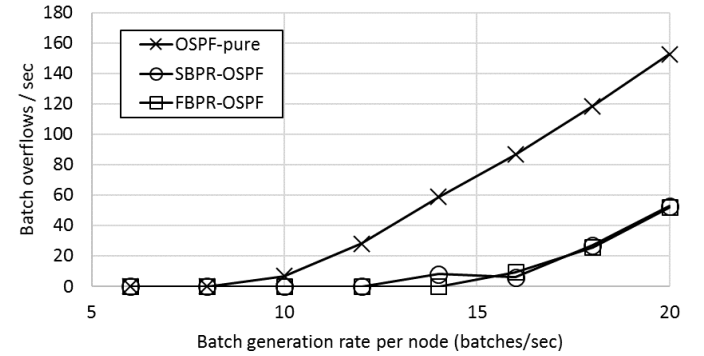
When enabled, the BPR-derived routing rules handle the enqueued batches in a LIFO manner, as advised in [19]. This holds for both SBPR and FBPR in the ensuing comparisons. The Open-Shortest-Path-First (OSPF) approach is used as the underlying DVR routing scheme in all applicable cases. Finally, while *FBPR-OSPF* is loop-free due to the described, hop-based filtering at line 5 of Algorithm 1, the routing rules proposed by *SBPR-OSPF* may create loops. Therefore, a pairwise check is performed among the nodes for the detection of loops. If one exists, the specific BPR-derived priority routing rules that caused it are filtered-out and are not forwarded to the nodes.

Figure 2 illustrates the performance of pure *OSPF* (no overlaid BPR), *SBPR-OSPF* and *FBPR-OSPF*, for varying network load. The x-axis corresponds to the number of batches generated at each node per second, G , which is uniform for all nodes ($G_{ij} = G, \forall i, j$). A load of 5 batches per second corresponds to $500MBps$ data generation rate. For a node being serviced by four outgoing channels of $2GBps$ each, this translates to a 1 : 16 channel over-subscription rate with regard to local users only. At $G = 20$ batches per second, the ratio rises to 1 : 4. The actuation period, T is set to $5sec$ and the alarm level is 20% of the buffer size. In terms of batch latency, Fig. 2a shows that the proposed *FBPR-OSPF* approach offers the best latency times, even over OSPF, until $G \approx 17$ batches/sec. At that point, OSPF-pure offers better latency, at the expense of an excessive traffic overflow rate (Fig. 2b). As expected, dropping much of the flowing traffic benefits the delivery times of the “surviving” traffic. However, all BPR-based schemes are able to sustain operation with a limited overflow rate, even under maximal load. In other words, the stability of the system is clearly increased with the use of the BPR class of routing schemes. This phenomenon is also evident from the throughput plot of Fig. 2c. OSPF-pure offers the worst performance, since it leads to queue build-up and high overflow rate. On the other hand, the proposed *FBPR-OSPF* offers significantly improved results. Nonetheless, *SBPR-OSPF* offers the maximum throughput in all cases. However, given its performance in term of latency, the superiority in raw throughput is clearly not useful and is owed to batches traveling via excessively long routes within the network [18].

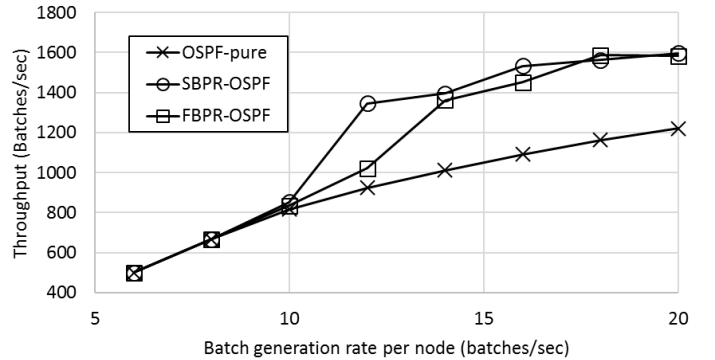
We proceed to study the benefits of endowing BPR with foresight. In Fig. 3, the batch generation rate per node is set randomly at $G_{ij} = G \pm v \cdot G$ (uniform distribution) where v is a percentage ranging from 10% to 50%. Notice that, in the previous experiment, FBPR and SBPR were equivalent from the aspect of foresight, due to the constant G values over all nodes. The alarm level is kept at 20% of the buffer size and T is varied from 5 to 15 sec. Each point in Fig. 3 is derived as the average over 50 simulation iterations. Since the goal of the comparison is to deduce the gains derived from foresight, perfect knowledge of G_{ij} is passed to *FBPR-OSPF*. Furthermore, for fairness reasons, the latency-favoring, hop-based node filtering of FBPR



(a) Achieved average batch delivery times.



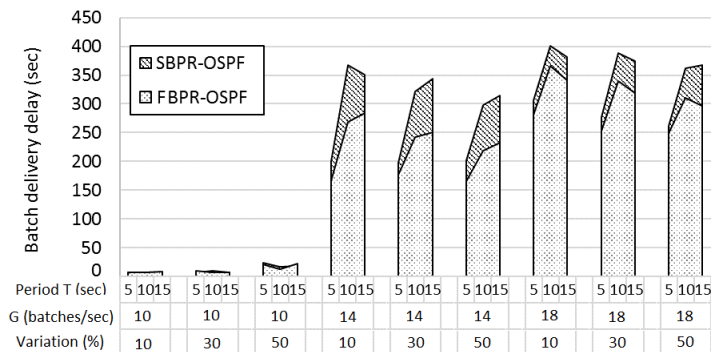
(b) Achieved average batch overflow rates.



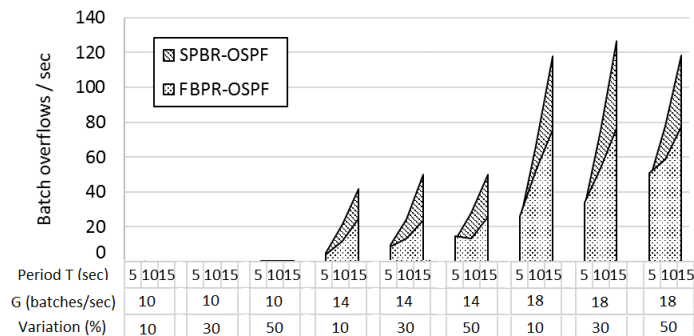
(c) Achieved average throughput.

Figure 2. The comparative performance of the Backpressure-based schemes and a standalone OSPF approach. The alarm level is 20% and actuation period $T = 5sec$.

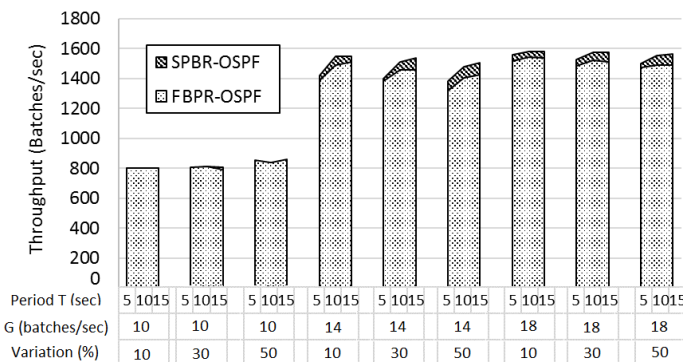
is discarded. (i.e. line 5 of Algorithm 1 considers all neighbors of node n). Thus, *FBPR-OSPF* drops any latency considerations that could have given an advantage over *SBPR-OSPF* from this aspect. The performance gains in batch latency and overflow rate are apparent in Fig. 3a and 3b respectively. In general, the bonus of foresight is significant as T increases, since the system can make more long-lived routing decisions. The gains are also accentuated for medium to high network loads, where BPR in general makes sense. The trade-off between latency and overflow rates is present in 3a and 3b as well. Finally, the throughput optimality continues to hold (Fig. 3c) with the slight difference being owed to the redundant data traveling produced by SBPR. In other words, having no foresight, SBPR takes decisions that distribute the network traffic slightly wider, but lead to higher latency and overflow rate in the future.



(a) Average batch delivery times.



(b) Average batch overflow rates.



(c) Average throughput.

Figure 3. The Foresight-enabled backpressure routing yields significant performance gains compared to the standard backpressure algorithm, while retaining the throughput-optimality trait.

VI. CONCLUSION

The present study brought Backpressure routing (BPR) and its benefits to the SDN-derived traffic engineering ecosystem. Its inherited benefits include throughput maximization and optimal stability under increased network load. The BPR and SDN combination can offer attractive, lightweight and centrally orchestrated routing solutions. Minimum cost, non-penetrative approaches could be the key for gradually encouraging cooperation between distrustful autonomous parties, with significant gains for the end-users. The presented approach can pave the way for a new class of lightweight traffic engineering schemes that require minimal commitment from the orchestrated network elements.

VII. ACKNOWLEDGEMENT

This work was funded by the EU project Net-Volution (EU338402) and the Research Committee of the Aristotle University of Thessaloniki.

REFERENCES

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [2] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven WAN," in *Proceedings of the ACM SIGCOMM conference*, 2013, pp. 15–26.
- [3] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *INFOCOM, 2013 Proceedings IEEE*, 2013, pp. 2211–2219.
- [4] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proceedings of the ACM SIGCOMM conference*, 2013, pp. 3–14.
- [5] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in *Proceedings of the first workshop on Hot topics in SDNs*, 2012, pp. 19–24.
- [6] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying NOX to the Datacenter," in *HotNets*, 2009.
- [7] M. McBride, M. Cohn, S. Deshpande, M. Kaushik, M. Mathews, and S. Nathan, "SDN Security Considerations in the Data Center," *Open Networking Foundation-ONF SOLUTION BRIEF*, 2013.
- [8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [9] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: fine grained traffic engineering for data centers," in *Proceedings of the seventh CONEXT conference*, 2011, p. 8.
- [10] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, 2010.
- [11] A. R. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1629–1637.
- [12] C. E. Hopps, "Analysis of an equal-cost multi-path algorithm," 2000.
- [13] R. Wojcik, J. Domzal, and Z. Dulinski, "Flow-Aware Multi-Topology Adaptive Routing," *IEEE Communications Letters*, vol. 18, no. 9, pp. 1539–1542, 2014.
- [14] J. Domzał, Z. Duliński, M. Kantor, J. Rząsa, R. Stankiewicz, K. Wajda, and R. Wójcik, "A survey on methods to provide multipath transmission in wired packet networks," *Comp. Networks*, vol. 77, pp. 18–41, 2015.
- [15] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks," *FNT in Networking (Foundations and Trends in Networking)*, vol. 1, no. 1, pp. 1–144, 2005.
- [16] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89–103, 2005.
- [17] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.
- [18] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On Combining Shortest-Path and Back-Pressure Routing Over Multihop Wireless Networks," *IEEE/ACM Trans. on Networking*, vol. 19, no. 3, pp. 841–854, 2011.
- [19] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFO-Backpressure Achieves Near-Optimal Utility-Delay Tradeoff," *IEEE/ACM Trans. on Networking*, vol. 21, no. 3, pp. 831–844, 2013.
- [20] H. Seferoglu and E. Modiano, "TCP-aware backpressure routing and scheduling," in *2014 Information Theory and Applications Workshop (ITA)*, pp. 1–9.
- [21] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A Low Cost Network Monitoring Framework for Software Defined Networks," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2014.
- [22] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "Flowsense: monitoring network utilization with zero measurement cost," in *Passive and Active Measurements*, 2013, pp. 31–41.
- [23] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: traffic matrix estimator for OpenFlow networks," in *Passive and Active Measurement*, 2010, pp. 201–210.
- [24] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, "Generalized Hessian matrix and second-order optimality conditions for problems with C 1,1 data," *Applied Mathematics & Optimization*, vol. 11, no. 1, pp. 43–56, 1984.
- [25] XJ Technologies, "The AnyLogic Simulator," 2013. [Online]. Available: <http://www.xjtek.com/anylogic/>