Fifty-second Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 1 - 3, 2014

Throughput Optimal Routing in Overlay Networks

Georgios S. Paschos and Eytan Modiano
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

***Abstract*—Maximum throughput requires path diversity enabled by bifurcating traffic at different network nodes. In this work, we consider a network where traffic bifurcation is allowed only at a subset of nodes called *routers*, while the rest nodes (called *forwarders*) cannot bifurcate traffic and hence only forward packets on specified paths. This implements an overlay network of routers where each overlay link corresponds to a path in the physical network. We study dynamic routing implemented at the overlay. We develop a queue-based policy, which is shown to be maximally stable (throughput optimal) for a restricted class of network scenarios where overlay links do not correspond to overlapping physical paths. Simulation results show that our policy yields better delay over dynamic policies that allow bifurcation at all nodes, such as the backpressure policy. Additionally, we provide a heuristic extension of our proposed overlay routing scheme for the unrestricted class of networks.**

I. Introduction

A common way to route data in communication networks is shortest path routing. Routing schemes using shortest path are *single-path*; they route all packets of a session through the same dedicated path. Although single-path schemes thrive because of their simplicity, they are in general throughput suboptimal. Maximizing network throughput requires *multi-path routing*, where the different paths are used to provide diversity [4].

When the network conditions are time-varying or when the session demands fluctuate unpredictably, it is required to balance the traffic over the available paths using a *dynamic routing* scheme which adapts to changes in an online fashion. In the past, schemes such as *backpressure* [13] have been proposed to discover multiple paths dynamically and mitigate the effects of network variability. Although backpressure is desirable in many applications, its practicality is limited by the fact that it requires all nodes in the network to make online routing decisions. Often it is the case that some network nodes have limited capabilities and cannot perform such actions. *In this paper we study dynamic routing when decisions can be made only at a subset of nodes, while the rest nodes use fixed single-path routing rules.*

Network overlays are frequently used to deploy new communication architectures in legacy networks [11]. To accomplish this, messages from the new technology are encapsulated in the legacy format, allowing the two methods to coexist in the legacy network. Nodes equipped with the new technology are then connected in a conceptual network overlay, Fig. 1. Prior works have considered the use of this methodology to introduce new routing capabilities in the Internet. For example, content providers use overlays to balance the traffic across different Internet paths and improve resilience and end-to-end performance [1], [12]. In our work we use a network overlay to introduce dynamic routing to a legacy network which operates based on single-path routing. Nodes that implement the overlay layer are called *routers* and are able to make online routing decisions, bifurcating traffic along different paths. The rest nodes, called *forwarders*, rely on a single-path routing protocol which is available to the physical network, see Fig. 1.

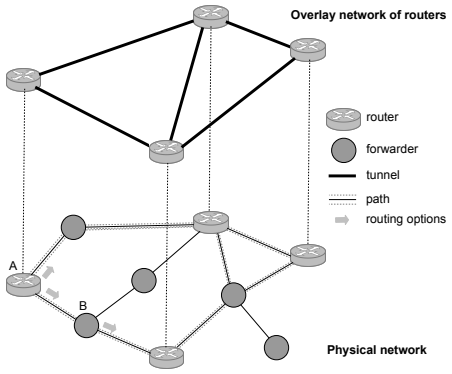

Fig. 1. Router A can bifurcate traffic while forwarder B only forwards the packets along a predetermined path. This paper studies dynamic routing in the overlay.

There are many applications of our overlay routing model. For networks with heterogeneous technologies, the overlay routers correspond to devices with extended capabilities, while the forwarders correspond to less capable devices. For example, to introduce dynamic routing in a network running a legacy routing protocol, it is possible to use Software Defined Networks to install dynamic routing functions on a subset of devices (the routers). In the paradigm of multi-owned networks, the forwarders are devices where the vendor has no administrative rights. For example consider a network that uses leased satellite links, where the forwarding rules may be pre-specified by the lease. In such heterogeneous scenarios, maximizing throughput by controlling only a fraction of nodes introduces a tremendous degree of flexibility.

In the physical network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ denote the set of routers with $\mathcal{V} \subseteq \mathcal{N}$. Also, denote the throughput region of this network with $\Lambda(\mathcal{V})$ [5].[1] Then, $\Lambda(\mathcal{N})$ is the throughput

This work was supported by NSF grant CNS-1217048, ONR grant N00014-12-1-0064, and ARO MURI grant W911NF-08-1-0238. The work of G. Paschos is supported by the WiNC project of the Action:Supporting Postdoctoral Researchers, funded by national and Community funds (European Social Fund).

[1] The definition of throughput region is given later; here it suffices to think of the set of feasible throughputs.

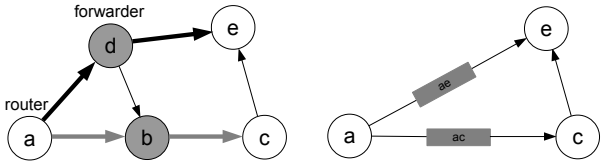

Fig. 2. (left) An example network of routers and forwarders, where routers are $\mathcal{V} = \{\mathsf{a},\mathsf{c},\mathsf{e}\}$. We indicate with bold arrows the shortest paths available to a by the single-path routing scheme of the physical network. (right) The equivalent overlay network of routers and tunnels.

of the network when all nodes are routers. We call this the full throughput of $\mathcal{G}$, and it can be achieved if all nodes run the backpressure policy [13]. Also, $\Lambda(\emptyset)$ is the throughput of a network consisting only of forwarders, which is equivalent to single-path throughput. Since increasing the number of routers increases path diversity, we generally have $\Lambda(\emptyset) \subseteq \Lambda(\mathcal{V}) \subseteq \Lambda(\mathcal{N})$. Prior work studies the necessary and sufficient conditions for router set $\mathcal{V}^*$ to guarantee full throughput, i.e., $\Lambda(\mathcal{V}^*) = \Lambda(\mathcal{N})$ [6]. The results of the study show that using a small percentage of routers (8%) is sufficient for full throughput in power-law random graphs–an accurate model of the Internet [9]. Although [6] characterizes the throughput region $\Lambda(\mathcal{V})$, a dynamic routing to achieve this performance is still unknown. For example, in the same work it is showcased that backpressure operating in the overlay is suboptimal. *In this work we fill this gap under a specific topological assumption explained in detail later. We study dynamic routing in the overlay network of routers and propose a control policy that achieves $\Lambda(\mathcal{V})$. Our work is the first to analytically study such a heterogeneous dynamic routing policy and prove its optimality.*

II. System Model

We consider a physical network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where the nodes are partitioned to routers $\mathcal{V}$ and forwarders $\mathcal{N} - \mathcal{V}$. The physical network has installed single-path routing rules, which we capture as follows. Every router $i \in \mathcal{V}$ is assigned an acyclic path p_{ij} to every other router $j \in \mathcal{V}$.[2] Fig. 2 (left) shows with bold arrows both paths assigned to router a, i.e., (a,d,e), and (a,b,c). Let P be the set of all such paths in the network.

A. The Overlay Network of Tunnels

We introduce the concept of *tunnels*. The tunnel $(i,j) \in \mathcal{E}$ corresponds to a path $p_{ij} \in P$ with end-points routers i, j and intermediate nodes forwarders. We then define the overlay network $\mathcal{G}_R = (\mathcal{V}, \mathcal{E})$ consisting of routers $\mathcal{V}$ and tunnels $\mathcal{E}$. Figure 2 (right) depicts the overlay network for the physical network in the left, assuming shortest path routing is used.

1) Topological Assumption: In this work we study the case of *non-overlapping tunnels*. Let $\mathcal{T}_{ij}$ be the set of all physical links of tunnel (i,j) with the exception of the first input link.

Definition 1 [Non-Overlapping Tunnels]: *An overlay network satisfies the non-overlapping tunnels condition if for any two tunnels $e_1 \neq e_2$ we have $\mathcal{T}_{e_1} \cap \mathcal{T}_{e_2} = \emptyset$.*

[2]The legacy routing protocol may provide paths between physical nodes as well, but we do not study them in this work.

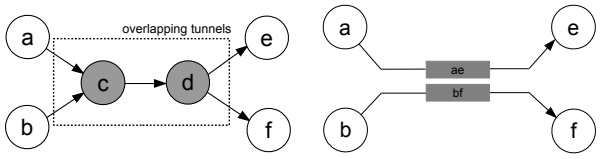

Fig. 3. An example with overlapping tunnels.

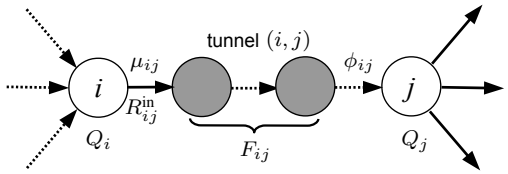

Fig. 4. The input of a tunnel is controllable (solid line) but the output is uncontrollable (dotted line).

Whether the condition is satisfied or not, depends on the network topology $\mathcal{G}$, the set of routers $\mathcal{V}$, and the set of paths P which altogether determine $\mathcal{T}_{ij}$, for all $i, j \in \mathcal{V}$. The network of Figure 2 satisfies the non-overlapping tunnels condition since each of the links (d,e), (b,c) belongs to exactly one tunnel. On the other hand, in the network of Figure 3 link (c,d) belongs to two tunnels, hence the condition is not satisfied.

When tunnels overlap, packets belonging to different tunnels compete for service at the forwarders, which further complicates the analysis. Our analytical results focus exclusively on the non-overlapping tunnels case which still constitutes an interesting and difficult problem. However, in the simulation section we heuristically extend our proposed policy to apply to general networks with overlapping tunnels and showcase that the extended policy has near-optimal performance.

B. Overlay Queueing Model

The overlay network admits a set of sessions $\mathcal{C}$, where each session has a unique router destination, but possibly multiple router sources. Time is slotted; at the end of time slot t, $A_i^c(t) \leq A_{\max}$ packets of session $c \in \mathcal{C}$ arrive exogenously at router i, where $A_{\max}$ is a positive constant. [3] $A_i^c(t)$ are i.i.d. over slots, independent across sessions and sources, with mean λ_i^c.

For every tunnel (i,j), a routing policy π chooses the routing function $\mu_{ij}^c(t,\pi)$ in slot t which determines *the number of session c packets* to be routed from router i into the tunnel. Additionally, we denote with $\phi_{ij}^c(t)$ the actual number of session c packets that exit the tunnel in slot t. For a visual association of $\mu_{ij}^c(t,\pi)$ and $\phi_{ij}^c(t)$ to the tunnel links see Figure 4. Note that $\mu_{ij}^c(t,\pi)$ is decided by router i while $\phi_{ij}^c(t)$ is uncontrollable.

Let the sets $\text{In}(i), \text{Out}(i)$ represent the incoming and outgoing neighbors of router i on $\mathcal{G}_R$. Packets of session c are stored at router i in a *router queue*. Its backlog $Q_i^c(t)$ evolves

[3]Note that we focus exclusively on routing at the overlay layer. Thus $A_i^c(t)$ are defined at overlay router nodes.

according to the following equation

$$Q_i^c(t+1) = \Big(Q_i^c(t) - \underbrace{\sum_{b\in \text{Out}(i)} \mu_{ib}^c(t,\pi)}_{\text{departures}}\Big)^+ + \underbrace{\sum_{a\in \text{In}(i)} \phi_{ai}^c(t) + A_i^c(t)}_{\text{arrivals}}, \quad (1)$$

where we use $(.)^+ \triangleq \max\{.,0\}$ since there might not be enough packets to transmit.

On tunnel (i,j) we collect all packets into one *tunnel queue* $F_{ij}(t)$ whose evolution satisfies

$$F_{ij}(t+1) \le F_{ij}(t) - \underbrace{\sum_c \phi_{ij}^c(t)}_{\text{departures}} + \underbrace{\sum_c \mu_{ij}^c(t,\pi)}_{\text{arrivals}}, \ \forall (i,j)\in\mathcal{E}. \quad (2)$$

The packets that actually arrive at $F_{ij}(t)$ might be less than $\sum_c \mu_{ij}^c(t,\pi)$, hence the inequality (2). We remark that $F_{ij}(t)$ is the total number of packets in flight on the tunnel (i,j). Physically these packets are stored at different forwarders along the tunnel. We only keep track of the sum of these physical backlogs since, as we will show shortly, this is sufficient to achieve maximum throughput.

Above (1) assumes that all incoming traffic at router i arrives either from tunnels, or exogenously. It is possible, however, to have an incoming neighbor router k such that (k,i) is a physical link, a case we purposely omitted in order to avoid further complexity in the exposition. The optimal policy for this case can be obtained from our proposed policy by setting the corresponding tunnel queue backlog to zero, $F_{ki}(t) = 0$.

C. Forwarder Scheduling Inside Tunnels

We assume that inside tunnels packets are forwarded in a *work-conserving* fashion, i.e., a forwarder does not idle unless there is nothing to send. Due to work-conservation and the assumption of non-overlapping tunnels, a tunnel with "sufficiently many" packets has instantaneous output equal to its bottleneck capacity. Denote by M_{ij} the number of forwarders associated with tunnel (i,j). Let $R_{ij}^{\max}$ be the greatest capacity among all physical links associated with tunnel (i,j) and $R_{ij}^{\min}$ the smallest, also let

$$T_0 \triangleq \max_{(i,j)\in\mathcal{E}} \left[M_{ij} R_{ij}^{\min} + \frac{M_{ij}(M_{ij}-1)}{2} R_{ij}^{\max} \right]. \quad (3)$$

LEMMA 1 [OUTPUT OF A LOADED TUNNEL]: *Under any control policy $\pi \in \Pi$, suppose that in time slot t the total tunnel backlog satisfies $F_{ij}(t) > T_0$, for some $(i,j)\in\mathcal{E}$, where T_0 is defined in* (3)*. The instantaneous output of the tunnel satisfies*

$$\sum_c \phi_{ij}^c(t) = R_{ij}^{\min}. \quad (4)$$

Proof: The proof is provided in the Appendix. ∎

Lemma 1 is a path-wise statement saying that the tunnel output is equal to the tunnel bottleneck capacity in every time slot that the tunnel backlog exceeds T_0.

Notably we haven't discussed yet how the forwarders choose to prioritize packets from different sessions. Based on Lemma 1 and the results that follow, we will establish that independent of the choice of session scheduling policy, there exists a routing policy that maximizes throughput. Furthermore, we demonstrate by simulations that different forwarding scheduling policies result in the same average delay performance under our proposed routing. Hence, in this paper forwarders are allowed to use any work-conserving session scheduling, such as FIFO, Round Robin or even strict priorities among sessions.

III. DYNAMIC ROUTING PROBLEM FORMULATION

A choice for the routing function $\mu_{ij}^c(t,\pi)$ is considered permissible if it satisfies in every slot the corresponding capacity constraint $\sum_c \mu_{ij}^c(t,\pi) \le R_{ij}^{\text{in}}$, where R_{ij}^{in} denotes the capacity of the input physical link of tunnel (i,j), see Fig. 4. In every time slot, a control policy π determines the routing functions $\big(\mu_{ij}^c(t,\pi)\big)$ at every router. Let Π be the class of all permissible control policies, i.e., the policies whose sequence of decisions consists of permissible routing functions.

We want to keep the backlogs small in order to guarantee that the throughput is equal to the arrivals. To keep track of this we define the stability criterion adopted from [5].

DEFINITION 2 [SYSTEM STABILITY]: *A queue with backlog $X(t)$ is stable under policy π if*

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[X(t)] < \infty.$$

The overlay network is stable if all router $(Q_i^c(t))$ and tunnel queues $(F_{ij}(t))$ are stable.

The *throughput region* $\Lambda(\mathcal{V})$ of class Π is defined to be (the closure of) the set of $\boldsymbol{\lambda} = (\lambda_i^c)$ for which there exists a policy $\pi \in \Pi$ such that the system is stable. Avoiding technical jargon, the throughput region includes all achievable throughputs when implementing dynamic routing in the overlay. Recall that throughput depends on the actual selection of routers $\mathcal{V}$, and that for $\mathcal{V} \subset \mathcal{N}$ it may be the case that the achievable throughput may be less than the full throughput of $\mathcal{G}$, i.e., $\Lambda(\mathcal{V}) \subset \Lambda(\mathcal{N})$. Therefore it is important to clarify that in this work we assume that $\mathcal{V}$ is fixed and we seek to find a policy that is stable for any $\boldsymbol{\lambda} \in \Lambda(\mathcal{V})$, i.e., a policy that is *maximally stable*. Such a policy is also called in the literature "throughput optimal".

A. Characterization of Throughput Region of Class Π

The throughput region $\Lambda(\mathcal{V})$ can be characterized as the closure of the set of matrices $\boldsymbol{\lambda} = (\lambda_i^c)$ for which there exist nonnegative flow variables (f_{ij}^c) such that

$$\lambda_i^c + \sum_{a\in\mathcal{V}} f_{ai}^c < \sum_{b\in\mathcal{V}} f_{ib}^c, \qquad \text{for all } i\in\mathcal{V}, c\in\mathcal{C} \quad (5)$$

$$\sum_c f_{ij}^c < R_{ij}^{\min}, \qquad \text{for all } (i,j),\in\mathcal{E}, \quad (6)$$

where (5) are flow conservation inequalities at routers, (6) are capacity constraints on tunnels, and recall that $R_{ij}^{\min}$ is

the bottleneck capacity in the tunnel (i,j). We write

$$\Lambda(\mathcal{V}) = \text{Cl}\{\boldsymbol{\lambda} \mid \boldsymbol{f} \geq \mathbf{0}, \text{ and (5)-(6) hold}\}.$$

Note, that the conditions for the stability region $\Lambda(\mathcal{V})$ are the same with the conditions for full throughput $\Lambda(\mathcal{N})$ [5], with the difference that the flow variables are defined on the network of routers $\mathcal{G}_R$ instead of $\mathcal{G}$. Indeed the proof that (5)-(6) are necessary and sufficient for stability may be obtained by considering a virtual network where every tunnel is replaced by a virtual link.

Controlling this system in a dynamic fashion amounts to finding a routing policy $\pi^* \in \Pi$ which stabilizes the system for any $\boldsymbol{\lambda} \in \Lambda(\mathcal{V})$. *Finding such a policy in the overlay differs significantly from the case of a physical network*, since physical links support immediate transmissions while overlay links are work-conserving tandem queues which induce queueing delays.

IV. The Proposed Routing Policy

As discussed in [6], using backpressure in the overlay may result in poor throughput performance. In this section we propose the Threshold-based Backpressure (BP-T) Policy, a distributed policy which performs online decisions in the overlay. BP-T is designed to operate the tunnel backlogs close to a threshold. This is a delicate balance whereby the tunnel output works efficiently (by Lemma 1) while at the same time the number of packets in the tunnel are upper bounded.

Consider the threshold

$$T = T_0 + \max_{(i,j)} R_{ij}^{\text{in}}, \tag{7}$$

where T_0 is defined in (3) and R_{ij}^{in} is the capacity of input physical link of tunnel (i,j) and thus also the maximum increase of the tunnel backlog in one slot. Define the condition:

$$F_{ij}(t) \leq T. \tag{8}$$

The reason we use this threshold is that if (8) is false, it follows that both $F_{ij}(t) > T_0$ and $F_{ij}(t-1) > T_0$, and hence we can apply Lemma 1 to both slots t and $t-1$. This is used in the proof of the main result.

Threshold-based Backpressure (BP-T) Policy

At each time slot t and tunnel (i,j), let

$$c_{ij}^* \in \arg\max_{c \in \mathcal{C}} Q_i^c(t) - Q_j^c(t),$$

be a session that maximizes the differential backlog between routers i,j, ties resolved arbitrarily. Then route into that tunnel

$$\mu_{ij}^{c_{ij}^*}(t, \text{TB}) = \begin{cases} R_{ij}^{\text{in}} & \text{if } Q_i^{c_{ij}^*}(t) > Q_j^{c_{ij}^*}(t) \\ & \text{AND } (8) \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and $\mu_{ij}^c(t, \text{BP-T}) = 0, \ \forall c \neq c_{ij}^*$. Recall, that R_{ij}^{in} denotes the capacity of input physical link of tunnel (i,j). [4]

[4] If the there are not enough packets to transmit, i.e., $\mu_{ij}^{c_{ij}^*}(t) > Q_i^{c_{ij}^*}(t)$, then we fill the transmissions with dummy non-informative packets.

BP-T is similar to applying backpressure in the overlay, with the striking difference that *no packet is transmitted to a tunnel* if condition (8) is not satisfied. Therefore the total tunnel backlog is limited to at most T plus the maximum number of packets that may enter the tunnel in one slot. Formally we have

Lemma 2 [Deterministic bounds of $F_{ij}(t)$ under BP-T]: *Assume that the system starts empty and is operated under BP-T. Then the tunnel backlogs $(F_{ij}(t))$ are uniformly bounded above by*

$$F^{\max} \triangleq T + R_{\max}. \tag{10}$$

Proof: Follows from (8) and (9). ■

This shows that our policy does not allow the tunnel backlogs to grow beyond $F^{\max}$. To show that our policy efficiently routes the packets is much more involved. It is included in the proof of the following main result.

Theorem 3: *[Maximal Stability of BP-T] Consider an overlay network where underlay forwarding nodes use any work-conserving policy to schedule packets over predetermined paths, and the tunnels are non-overlapping.*

The BP-T policy is maximally stable:

$$\Lambda^{\text{BP-T}}(\mathcal{V}) \supseteq \Lambda^{\pi}(\mathcal{V}), \text{ for all } \pi \in \Pi.$$

Proof: The proof is is based on a novel K-slot Lyapunov drift analysis and due to space limitations is given in [10]. ■

BP-T is a distributed policy since it utilizes only local queue information and the capacity of the incident links, while it is agnostic to arrivals, or capacities of remote links, e.g. note that the decision does not depend on the capacity of the bottleneck link $R_{ij}^{\min}$.

A very simple distributed protocol can be used to allow overlay nodes to learn the tunnel backlogs. Specifically $F_{ij}(t)$ can be estimated at node i using an acknowledgement scheme, whereby j periodically informs i of how many packets have been received so far. In practice, the router nodes obtain a delayed estimate $\tilde{F}_{ij}(t)$. However, using the concepts in [7]-p.85, it is possible to show that such estimates do not hurt the efficiency of the scheme.

V. Simulation Study

In this section we perform extensive simulations to:

(i) showcase the maximal stability of BP-T and compare its throughput performance to other routing policies,
(ii) examine the impact of different forwarding scheduling policies (FIFO, HLPSS, Strict Priority, LQF) on throughput and delay of BP-T,
(iii) demonstrate that BP-T has good delay performance, and
(iv) study the extension of BP-T to the case of overlapping tunnels.

First we present dynamic routing policies from the literature against which we will compare BP-T.

Backpressure in the overlay (BP-O): For every tunnel $(i,j) \in \mathcal{E}$ define

$$c_{ij}^* \in \arg\max_{c \in \mathcal{C}} Q_i^c(t) - Q_j^c(t),$$

ties solved arbitrarily. Then choose $\mu_{ij}^c(t, \text{BP-O}) = 0, c \neq c_{ij}^*$ and

$$\mu_{ij}^{c_{ij}^*}(t, \text{BP-O}) = \begin{cases} R_{ij}^{\text{in}} & \text{if } Q_i^{c_{ij}^*}(t) > Q_j^{c_{ij}^*}(t) \\ 0 & \text{otherwise.} \end{cases}$$

This corresponds to backpressure applied only to routers $\mathcal{V}$, which is admissible in our system, BP-O $\in \Pi$.

Backpressure in the physical network (BP): For every physical link $(m, n) \in \mathcal{L}$ define

$$c_{mn}^* \in \arg\max_{c \in \mathcal{C}} Q_m^c(t) - Q_n^c(t)$$

ties solved arbitrarily. Then choose $\mu_{mn}^c(t, \text{BP}) = 0, c \neq c_{mn}^*$ and

$$\mu_{mn}^{c_{mn}^*}(t, \text{BP}) = \begin{cases} R_{mn} & \text{if } Q_m^{c_{mn}^*}(t) > Q_n^{c_{mn}^*}(t) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

This is the classical backpressure from [13], applied to all nodes $\mathcal{N}$ in the network, and thus it is not admissible in the overlay, BP $\notin \Pi$, whenever $\mathcal{V} \subset \mathcal{N}$. Since this policy achieves the full throughput $\Lambda(\mathcal{N})$, we use it as a throughput benchmark.

Backpressure Enhanced with Shortest Paths Bias (BP-SP): For every node-session pair (m, c) define the hop count from m to the destination of c as h_n^c. For every physical link $(m, n) \in \mathcal{L}$ define

$$c_{mn}^* \in \arg\max_{c \in \mathcal{C}} Q_m^c(t) - Q_n^c(t) + h_m^c - h_n^c.$$

ties solved arbitrarily. Then choose $\mu_{mn}^c(t, \text{BP-SP})$ according to (11). This policy was proposed by [8] to reduce delays. When the congestion is small, the shortest path bias introduced by the hop count difference leads the packets directly to the destination without going through cycles or longer paths. Such a policy requires control at every node, and thus it is not admissible in the overlay, BP-SP $\notin \Pi$, whenever $\mathcal{V} \subset \mathcal{N}$. Since, however, it is known to achieve $\Lambda(\mathcal{N})$ and to outperform BP in terms of delay, it is useful for throughput and delay comparisons.

A. Showcasing Maximal Stability

Consider the network of Figure 5 (left), and define two sessions sourced at a; session 1 destined to e and session 2 to c. We assume that $R_{\text{ab}} = 2$ and all the other link capacities are unit as shown in the Figure. We choose R_{ab} in this way to make the routing decisions of session 1 more difficult. We show the full throughput region $\Lambda(\mathcal{N})$ achieved by BP, BP-SP which however are not admissible in the overlay. Then we experiment with BP-T, BP-O and we also show the throughput of plain Shortest Path routing. For BP-T, according to example settings and (7) it is $T_0 = 2$; we choose $T = 6$.

Since the example satisfies the non-overlapping tunnel condition, by Theorem 3 our policy achieves $\Lambda(\mathcal{V})$. This is verified in the simulations, see Figure 5 (right). From the figure we can conclude that for this example we have $\Lambda(\mathcal{V}) = \Lambda(\mathcal{N})$, although $\mathcal{V} \subset \mathcal{N}$. This is consistent to the findings of [6]. From the same Figure we see that both

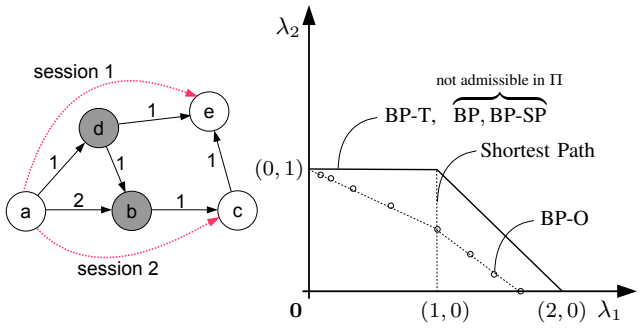

Fig. 5. Throughput comparison: (left) Example under study. (right) Throughput achieved by {BP-T, BP-O, Shortest Path} $\subset \Pi$ and BP, BP-SP $\notin \Pi$.

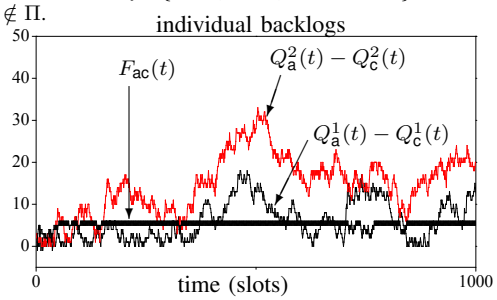

Fig. 6. Sample path evolution of the system under BP-T, $\lambda_1 = \lambda_2 = .97$.

backpressure in the overlay BP-O and Shortest Path achieve only a fraction of $\Lambda(\mathcal{V})$, and hence they are not maximally stable. For BP-O, we have loss of throughput when both sessions compete for traffic, in which case BP-O fails to consider congestion information from the tunnel ac and therefore allocates this tunnel's resources wrongly to the two sessions. For Shortest Path, it is clear that each session uses only its own dedicated shortest path and hence the loss of throughput is due to no path diversity.

To understand why BP-T works, we examine a sample path evolution of this system under BP-T for the case where $\lambda_1 = \lambda_2 = 0.97$, which is one of the most challenging scenarios. For stability, session 1 must use its dedicated path (a,d,e), and send almost no traffic through tunnel ac. Focusing on the tunnel ac, Figure 6 shows the differential backlogs per session $Q_{\text{a}}^c(t) - Q_{\text{c}}^c(t)$ and the corresponding tunnel backlog $F_{\text{ac}}(t)$ for a sample path of the system evolution. In most time slots a is congested, which is indicated by high differential backlogs. In such slots, the tunnel has more than 1 packet, which guarantees by Lemma 1 that it outputs packets at highest possible rate, hence the tunnel is correctly utilized. Recall that when the tunnel is full ($F_{\text{ac}}(t) > T$=6) no new packets are inserted to the tunnel preventing it from exceeding $F_{\max}$. Observe that the differential backlog of session 2 always dominates the session 1 counterpart, and hence whenever a tunnel is again ready for a new packet insertion, session 2 will be prioritized for transmission according to (9). Therefore, the proportion of session 2 packets in this tunnel is close to 100%, which is the correct allocation of the tunnel resources to sessions for this case.

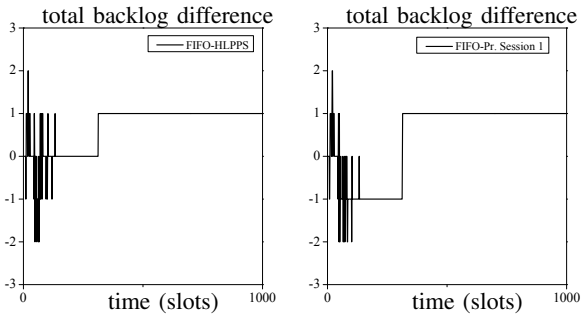

Fig. 7. Sample path difference in total system backlog, between different underlay forwarding policies: (left) difference between FIFO and HLPPS, (right) difference between FIFO and Strict Priority to session 1.

λ	FIFO	HLPPS	LQF	Priority Session 1
0.8	7.523	7.517	7.522	7.534
0.85	9.529	9.505	9.529	9.541
0.9	13.240	13.245	13.193	13.238
0.95	23.850	23.887	23.899	23.893
0.99	98.738	98.605	98.755	98.624

TABLE I
AVERAGE DELAY PERFORMANCE OF BP-T UNDER DIFFERENT UNDERLAY FORWARDING POLICIES.

B. Insensitivity to Forwarding Scheduling

At every forwarder node there is a packet scheduling decision to be made, to choose how many packets per session should be forwarded in the next slot. Although by assumption we require the forwarding policy to be work-conserving, our results do not restrict the scheduling policy any further. In particular, our analysis only depends on $\sum_c \phi_{ij}^c(t)$ and hence it is insensitive to the chosen discipline.

Here we simulate the operation of BP-T with different forwarding policies, in particular with First-In First-Out (FIFO), Head of Line Proportional Processor Sharing (HLPPS), Strict Priority and Longest Queue First (LQF), where HLPPS refers to serving sessions proportionally to their queue backlogs [2], and LQF refers to giving priority to the session with the longest queue. Figure 7 shows sample path differences for several forwarding disciplines on the example of the previous section, while Table I compares the average delay performance for different arrival rates. Independent of the discipline used, the average total number of packets in the system is approximately the same. Therefore, while our theorem states that the forwarding policy does not affect BP-T throughput, simulations additionally show that the delay is also the same.

C. Delay Comparison

We simulate the delay of different routing policies, comparing the performance of BP-T and BP-O overlay policies, as well as BP and BP-SP which are not admissible in the overlay. We experiment for $\lambda_1 = \lambda_2 = \lambda$, and we plot the average total backlogs in the system for two example networks shown to the left of each plot.

In Fig. 8 BP-O fails to detect congestion in the tunnel ac and consequently delay increases for $\lambda > 0.7$. We observe that BP-T outperforms BP and BP-O, and performs similarly to BP-SP. This relates to avoidance of cycles at low loads by use of shortest paths, see [5]. In particular, BP-SP achieves

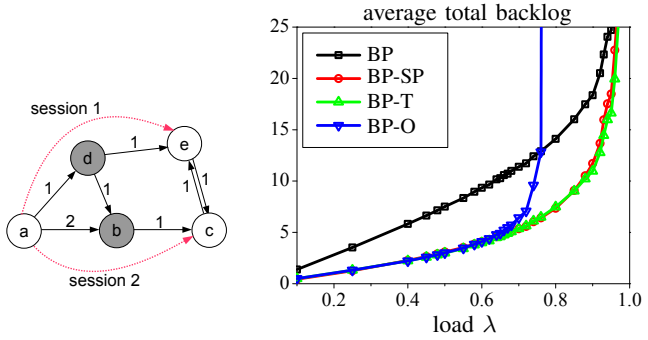

Fig. 8. Delay Comparison: (left) Example under study. (right) Average total backlog per offered load when $\lambda_1 = \lambda_2 = \lambda$.

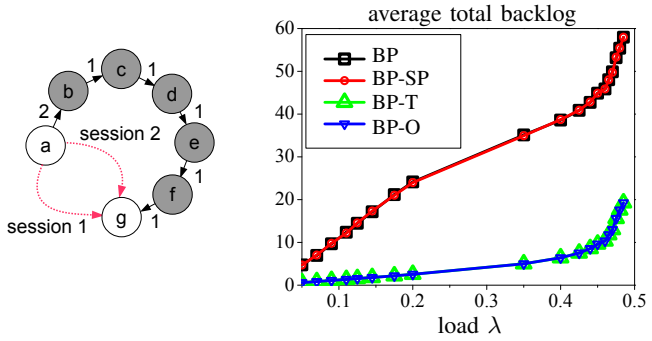

Fig. 9. Delay Comparison: (left) Example under study. (right) Average total backlog per offered load when $\lambda_1 = \lambda_2 = \lambda$.

this by means of hop count bias, while BP-T using the tunnels. *A remarkable fact is that BP-T applies control only at the overlay nodes and outperforms in terms of delay BP which controls all physical nodes in the network.*

In Fig. 9 we study queues in tandem, in which case all policies have maximum throughput since there is a unique path through which all the packets travel. We choose this scenario to demonstrate another reason why BP-T has good delay performance. The delay of backpressure increases quadratically to the number of network nodes because of maintaining equal backlog differences across all neighbors [3]. In the case of BP-T, as well as any other admissible overlay policy like BP-O, the backlogs increase with the number of routers. Thus, when $|\mathcal{V}| < |\mathcal{N}|$ we obtain a delay gain by applying control only at routers. Fig. 9 showcases exactly this delay gain that BP-T and BP-O have versus BP and BP-SP.

We conclude that BP-T has very good delay performance which is attributed to two main reasons:

1) When traffic load is low, the majority of the packets follow shortest paths. The number of packets going in cycles is significantly reduced.
2) Since there is no need for congestion feedback within the tunnels, the backlog buildup is not proportional to the number of network nodes but to the number of routers.

D. Applying our Policy to Overlapping Tunnels

Next we extend BP-T to networks with overlapping tunnels, see the example in Fig. 10 (left). In this context Theorem

3 does not apply and we have no guarantees that BP-T is maximally stable. The key to achieving maximum throughput is to correctly balance the ratio of traffic from each session injected into the overlapping tunnels. For the network to be stable with load $(.9, .9)$, a policy needs to direct most of the traffic of session 1 through the dedicated link (a, e), or equivalently to allocate $\mu^1_{\mathsf{ac}}(t) = 0$. Since node e is the destination of session 1, and hence $Q^1_{\mathsf{e}}(t) = 0$, we need to relate this routing decision to the congestion in the tunnel.

To make this work, we introduce the following extension. Instead of conditioning transmissions on router differential backlog $Q_i^{c^*_{ij}}(t) > Q_j^{c^*_{ij}}(t)$ as in BP-T, we use the condition $Q_i^{c^*_{ij}}(t) > Q_j^{c^*_{ij}}(t) + F_{ij}(t)$. Intuitively, we expect a non-congested node to have a small backlog and thus avoid sending packets over a congested tunnel. The new policy is called BP-T2. It can be proven that BP-T2 is maximally stable for non-overlapping tunnels. Although we do not have a proof for the case of overlapping tunnels, the simulation results show that by choosing T to be large BP-T2 achieves maximum throughput.

BP-T2 for Overlapping Tunnels

Fix a T to satisfy eq. (7), and recall condition (8):

$$F_{ij}(t) < T.$$

In slot t for tunnel (i, j) let

$$c^*_{ij} \in \arg\max_{c \in \mathcal{C}} Q_i^c(t) - Q_j^c(t),$$

be a session that maximizes the differential backlog between router i, j, ties resolved arbitrarily. Then route into tunnel (i, j)

$$\mu_{ij}^{c^*_{ij}}(t, \text{TB}) = \begin{cases} R^{\text{in}}_{ij} & \text{if } Q_i^{c^*_{ij}}(t) > Q_j^{c^*_{ij}}(t) + F_{ij}(t) \\ & \text{AND } (8) \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

and $\mu^c_{ij}(t, \text{BP-T}) = 0,\ \forall c \neq c^*_{ij}$. R^{in}_{ij} denotes the capacity of physical link that connects router i to the tunnel (i, j).

Figure 10 shows the results from an experiment where $T = 10$, $\lambda_1 = \lambda_2 = \lambda$, and we vary λ. BP-T2 achieves full throughput and similar delay to BP-SP, doing strictly better than BP-O, BP. To understand how BP-T2 works, consider the sample path evolution (Fig. 11), where $Q^1_{\mathsf{a}}(t) - Q^1_{\mathsf{e}}(t), Q^2_{\mathsf{b}}(t) - Q^2_{\mathsf{f}}(t), F_{\mathsf{ae}}(t)$ are shown. Most of the time we have $Q^1_{\mathsf{a}}(t) - Q^1_{\mathsf{e}}(t) < 10$, thus by the choice of $T = 10$ and the condition used in (12), session 1 rarely gets the opportunity to transmit packets to the overlapping tunnels. As T increases session 1 will get fewer and fewer opportunities, hence BP-T2 behavior will approximate the optimal. In Fig 11 (right) we plot the average total backlog for different values of T. As T increases, the performance at high loads improves.

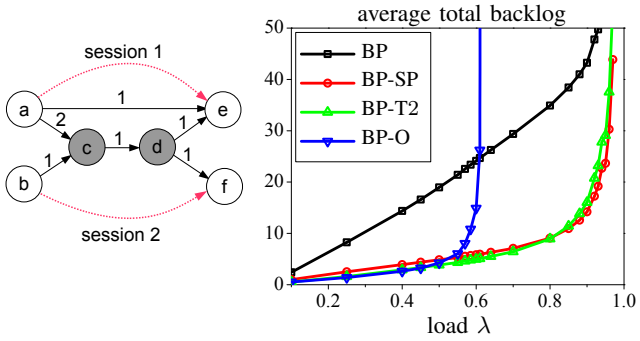

Fig. 10. Overlapping Tunnels: (left) Example under study. (right) Average total backlog per offered load when $\lambda_1 = \lambda_2 = \lambda$.

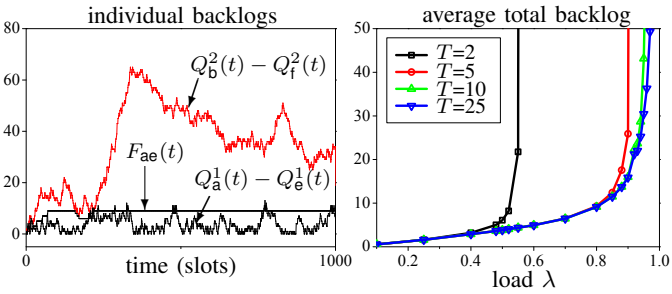

Fig. 11. (left) System evolution (one sample path) for $\lambda_1 = \lambda_2 = .97$, $T = 10$. (right) Average total backlog per offered load when $\lambda_1 = \lambda_2 = \lambda$.

VI. Conclusions

In this paper we propose a backpressure extension which can be applied in overlay networks. From prior work, we know that if the overlay is designed wisely, it can match the throughput of the physical network [6]. Our contribution is to prove that the maximum overlay throughput can be achieved by means of dynamic routing. Moreover, we show that our proposed scheme BP-T makes the best of both worlds (a) efficiently choosing the paths in online fashion adapting to network variability and (b) keeping average delay small avoiding the known inefficiencies of the legacy backpressure scheme.

Important future work involves the mathematical analysis of the overlapping tunnels case and the consideration of wireless transmissions. In both cases Lemma 1 does not hold due to correlation of routing decisions at routers with scheduling at forwarders.

References

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proc. ACM SOSP*, Oct. 2001.

[2] Maury Bramson. Convergence to equilibria for fluid models of head-of-the-line proportional processor sharing queueing networks. *Queueing Systems*, 23(1-4):1–26, 1996.

[3] L. Bui, R. Srikant, and A. Stolyar. Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing. In *Proc. IEEE INFOCOM*, April 2009.

[4] L.R. Ford and D.R. Fulkerson. Flows in networks. In *Princeton universtiy Press*, 1962.

[5] L. Georgiadis, M. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1:1–147, 2006.

[6] N. M. Jones, G. S. Paschos, B. Shrader, and E. Modiano. An overlay architecture for throughput optimal multipath routing. In *Proc. of ACM Mobihoc*, 2014.

[7] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
[8] Michael J. Neely, Eytan Modiano, and Charles E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23:89–103, 2005.
[9] M. E. J Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.
[10] G. S. Paschos and E. Modiano. Dynamic routing in overlay networks. Technical report, arXiv:1409.1739, 2014.
[11] L. L. Peterson and B. S. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition, 2007.
[12] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain. *Overlay Networks: An Akamai Perspective*. John Wiley & Sons, 2014.
[13] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936–1948, 1992.

APPENDIX

LEMMA 1 [OUTPUT OF A LOADED TUNNEL]: *Under any control policy $\pi \in \Pi$, suppose that in time slot t the total tunnel backlog satisfies $F_{ij}(t) > T_0$, for some $(i,j) \in \mathcal{E}$, where T_0 is defined in* (3)*. The instantaneous output of the tunnel satisfies*

$$\sum_c \phi_{ij}^c(t) = R_{ij}^{\min}. \tag{13}$$

Proof of Lemma 1: Consider a tunnel (i,j) which forwards packets, using an arbitrary work-conserving policy, over the path p_{ij} with M_{ij} underlay nodes. Renumber the nodes in the path in sequence they are visited by packets as $0, 1, \ldots, M_{ij}+1$, where 0 refers to i and $M_{ij}+1$ to j, hence

$$p_{ij} \triangleq \{0, 1, \ldots, M_{ij}, M_{ij}+1\}.$$

Since the statement is inherently related to packet forwarding internally in the tunnel (i,j), we will introduce some notation. Denote by $F_{ij}^k(t), k = 1, \ldots, M_{ij}$ the packets waiting at the k^{th} node at slot t, to be transmitted to the $k+1^{\text{th}}$, along tunnel $(i,j) \in \mathcal{V}$ (the packets may belong to different sessions). Clearly, it is $\sum_{k=1}^{M_{ij}} F_{ij}^k(t) = F_{ij}(t)$. Also, let $\phi_{ij}^{k,c}(t)$ be the actual number of session c packets that leave this backlog in slot t. For all $(i,j), k, c, t$, due to work-conservation we have

$$\sum_c \phi_{ij}^{k,c}(t) = \min\{R_k, F_{ij}^k(t)\}, \tag{14}$$

R_k denoting the capacity of the physical link connecting nodes $k, k+1$. Hence, $F_{ij}^k(t), k = 1, \ldots, M_{ij}$ evolve as

$$F_{ij}^k(t+1) = F_{ij}^k(t) - \sum_c \phi_{ij}^{k,c}(t) + \sum_c \phi_{ij}^{k-1,c}(t). \tag{15}$$

We begin the proof by showing that the instantaneous output of the tunnel cannot be larger than its bottleneck capacity, i.e.,

$$\sum_c \phi_{ij}^c(t) \le R_{ij}^{\min}. \tag{16}$$

If the bottleneck link is the last link on p_{ij} then (16) follows immediately from (14). Else, pick k such that $0 \le k < M_{ij}$

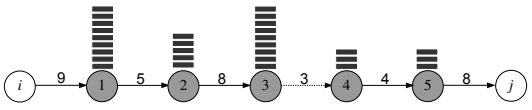

Fig. 12. An overloaded tunnel with bottleneck capacity $R_{ij}^{\min} = 3$.

and suppose $(k, k+1)$ is the bottleneck link. Then let us focus on the link $(k+1, k+2)$. For its input we have

$$\sum_c \phi_{ij}^{k,c}(t) \overset{(14)}{\le} R_k \triangleq R_{ij}^{\min}, \quad \text{for all } t$$

and for its output

$$\sum_c \phi_{ij}^{k+1,c}(t) = \min\{F_{ij}^{k+1}(t), R_{k+1}\},$$

where $R_{k+1} \ge R_k$. Starting the system empty, the backlog $F_{ij}^{k+1}(t)$ cannot grow larger than R_k since this is the maximum number of arriving packets in one slot and they are all served in the next slot. Hence, it is also $\sum_c \phi_{ij}^{k+1,c}(t) = F_{ij}^{k+1}(t) \le R_k$. By induction, the same is true for $F_{ij}^l(t), \phi_{ij}^l(t)$ for any $k < l \le M_{ij}$, and we get (16).

The remaining proof is by contradiction. Assume $\sum_c \phi_{ij}^c(t) < R_{ij}^{\min}$. Consider the physical link $(k, k+1)$ with $k = 2, \ldots, M_{ij}$. Using (15)

$$F_{ij}^k(t) < R_{ij}^{\min} \;\Rightarrow\; F_{ij}^{k-1}(t-1) < R_{ij}^{\min}. \tag{17}$$

To understand (17) note that if the RHS was false, by (14) we would have $\sum_c \phi_{ij}^{k-1,c}(t-1) \ge R_{ij}^{\min}$ and thus by (15) also $F_{ij}^k(t) \ge R_{ij}^{\min}$.

Since by the premise we have $\sum_c \phi_{ij}^{M_{ij},c}(t) \equiv \sum_c \phi_{ij}^c(t) < R_{ij}^{\min}$, applying (14) we deduce $F_{ij}^{M_{ij}}(t) < R_{ij}^{\min}$ from which applying (17) recursively we roll back in time and space to obtain

$$F_{ij}^k(t - M_{ij} + k) < R_{ij}^{\min}, \quad k = 1, \ldots, M_{ij}.$$

Since the maximum backlog increase at any node within one slot is $R_{ij}^{\max}$, we roll forward in time to get

$$F_{ij}^k(t) < R_{ij}^{\min} + (M_{ij} - k)R_{ij}^{\max}, \quad k = 1, \ldots, M_{ij}.$$

Summing up for all forwarders $k = 1, \ldots, M_{ij}$ we get

$$\begin{aligned} F_{ij}(t) = \sum_{k=1}^{M_{ij}} F_{ij}^k(t) &< \sum_{k=1}^{M_{ij}} \left[R_{ij}^{\min} + (M_{ij} - k)R_{ij}^{\max}\right] \\ &= M_{ij}R_{ij}^{\min} + \frac{M_{ij}(M_{ij}-1)}{2} R_{ij}^{\max} \overset{(3)}{=} T_0. \end{aligned} \tag{18}$$

which contradicts the premise of the lemma. ∎