LifeTime-Aware Backpressure—A New Delay-Enhanced Backpressure-Based Routing Protocol

Abdelbaset Kabou, Nadia Nouali-Taboudjemat, Soufiene Djahel, *Senior Member, IEEE*,
Saïd Yahiaoui, and Omar Nouali

Abstract—**Dynamic backpressure is a highly desirable family of routing protocols known for their attractive mathematical properties. However, these protocols suffer from a high end-to-end delay making them inefficient for real-time traffic with strict end-to-end delay requirements. In this paper, we address this issue by proposing a new adjustable and fully distributed backpressure-based scheme with low queue management complexity, named lifetime-aware backpressure. The novelty in the proposed scheme consists in introducing the urgency level as a new metric for service differentiation among the competing traffic flows in the network. Our scheme not just significantly improves the quality of service provided for real-time traffic with stringent end-to-end delay constraints, but interestingly protects also the flows with softer delay requirements from being totally starved. The proposed scheme has been evaluated and compared against other state-of-the-art routing protocol, using computer simulation, and the obtained results show its superiority in terms of the achieved end-to-end delay and throughput.**

Index Terms—**Backpressure (BP) routing, end-to-end delay, ns-3, quality of service (QoS), wireless networks.**

I. Introduction

NOWADAYS, real-time applications are gaining a rapidly growing popularity and being used in various domains ranging from military and aerospace industry to transportation and healthcare systems. In internet and multimedia, an unprecedented growth of real-time applications deployment is observed due to the emergence of voice over IP (VoIP) type communication paradigm, in addition to video conferencing and streaming. Such applications can be used for infotainment purposes as well as for more life-critical situations such as in military operations, disaster relief, and other mission critical scenarios [1].

Apart from the high throughput requirement, temporal aspects in these life-critical scenarios have a significant impact on the correctness of the whole system. It is, therefore, important for a rapidly deployed communication infrastructure to ensure a response within specified time constraints, often referred to as *deadlines*. In order to meet these stringent constraints, several networking challenges need to be solved. One of the major problems is the need for a robust resource management framework, starting with an efficient routing protocol.

The backpressure (BP) routing algorithms, which have received much attention from the research community in recent years [2], are basically a good candidate in such situations. With their appealing mathematical properties, these protocols are proven to be, theoretically, throughput optimal and offering as well, a very satisfactory stability level. First introduced by Tassiulas [3], such simple queue-differential-based scheduling and routing policy was shown to be able to stabilize the network under any feasible traffic rate vector [3]. However, even though it delivers maximum throughput by adapting itself to network conditions, there are several issues that have to be addressed before it can be widely deployed to transport traffic flows with strict end-to-end delay constraints.

As stated in the original paper [3], the BP algorithm assumes the existence of a central controller with a global view of the whole network, to perform complex computations at each time slot. Such requirements in addition to the computational complexity are too prohibitive in practice [4]. Moreover, this algorithm requires maintaining a queue for each potential destination at each node, which may limit its scalability to large networks due to the induced excessive overhead. Decreasing the computational complexity and proposing distributed schemes for the classical BP were the aim of a number of research works in recent years. For more details about these works, we refer the reader to [2], which presents an in-depth review of the most important recent results.

Besides the aforementioned issues, the BP algorithm is also suffering from a serious limitation. In certain conditions, this protocol is known for its long routing convergence times, and thus, its inefficiency in terms of quality of service (QoS) delivery, especially the end-to-end delay. While this is not problematic in the context of delay tolerant applications, it is unacceptable in the case of traffic with strict end-to-end delay requirements [5]. Although several attempts have been made recently to enhance the legacy BP performance in terms of the achieved end-to-end delay, none of them addressed the situation where multiple

Manuscript received April 25, 2017; revised July 27, 2017 and November 20, 2017; accepted December 22, 2017. *(Corresponding author: Abdelbaset Kabou.)*

A. Kabou is with the National Higher School of Computer Science, Algiers 16309, Algeria, and also with the Research Center on Scientific and Technical Information, Algiers 16306, Algeria (e-mail: a_kabou@esi.dz).

N. Nouali-Taboudjemat, S. Yahiaoui, and O. Nouali are with the Research Center on Scientific and Technical Information, Algiers 16306, Algeria (e-mail: nnouali@cerist.dz; syahiaoui@cerist.dz; onouali@cerist.dz).

S. Djahel is with the School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester M15 6BH, U.K. (e-mail: s.djahel@mmu.ac.uk).

Digital Object Identifier 10.1109/JSYST.2017.2789288

traffic flows, with different end-to-end delay requirements, are injected into the same network. In contrast, these works assume that the injected traffic flows exhibit similar characteristics, which is an unrealistic assumption.

Our main objective in this paper is to design a new BP-based scheme able to provide the previous temporal aspect in per-flow basis. Besides ensuring the best performance for applications with strict QoS requirements, the scheme must protect traffic flows with soft delay requirements from starvation (i.e., to receive zero or close-to-zero throughput). In the following, we summarize the main contributions of this work.

1) The consideration of more realistic assumptions regarding lower layers and the use of less complex queue management strategy, while proposing a new distributed BP-based scheme with enhanced end-to-end delay performance.
2) The routing layer of the proposed scheme is able to adjust the forwarding decision depending on the urgency level of each packet, in a way to prioritize traffic flows with strict end-to-end delay requirements.
3) The proposed scheme, while prioritizing these latter flows, is still able to avoid the starvation of traffic flows with soft requirements, by providing a minimum throughput for them, even in congested situations.

The remainder of this paper is organized as follows. We first present the main reasons behind the delay inefficiency of the BP and briefly review the related enhancements proposed in the literature. In Section III, we describe in detail the design aspects and basic components of our scheme. The simulation setup and performance evaluation are presented in Section IV. Finally, Section V concludes the paper and summaries the future perspectives.

II. BP Algorithms

In this section, we start with the system model and a brief description of how routing decisions are made in the classical BP protocol. Then, we present a review of the basic reasons behind the delay inefficiency of this protocol with discussion of the related works in the literature.

A. System Model

Let us consider a wireless multihop network described by a directed graph $G = (N, L)$, where N denotes the set of nodes and L refers to the set of links. Packets are injected at the source node and traverse multiple links to reach their destination via multihop communication. We assume that our system operates in a slotted time and that each link in the set L is denoted by (i, j) or l. The set of all per-destination traffic flows $f_{c\epsilon N}$ crossing the link (i, j) is represented by $F(i, j)$. For each traffic flow f_c crossing a node i a first-in first-out queue $Q_i^{f_c}$ is maintained by this node, as illustrated in the example shown in Fig. 1.

The maximum number of packets that a link (i, j) can transmit at one time slot is called the $link\ capacity$ and denoted as $\mu_{(i,j)}$. The set of all links capacities defines the so-called $network\ capacity\ vector\ c$ which, in turn, constitutes the basic element of the $network\ capacity\ region$ concept, denoted as Γ. Indeed, the collection of all possible network capacity vectors defines Γ. Table I recapitulates the main notations used throughout the rest of this paper.

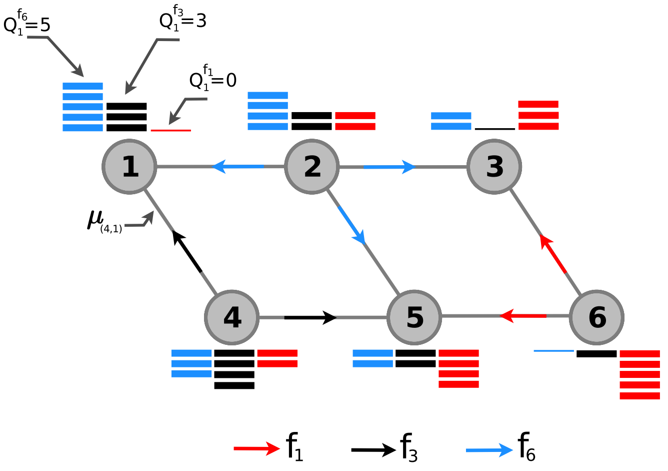


Fig. 1. Example of a wireless multihop network of six nodes, with a closeup on the queues at each node. Three traffic flows are injected: The red (f_1) destined to node 1, the black (f_3) destined to node 3, and the blue (f_6) destined to node 6.

TABLE I
Notations Summary

Symbol	Explanation
(N, L)	The set of nodes (N) and links (L) in the network.
(i, j)	A link between nodes i and j.
f_c	The traffic flow destined to the node c.
$F_{(i,j)}$	The set of traffic flows crossing the link (i, j).
$Q_i^{f_c}$	Local per-flow queue at the node i.
tag_p	The tag of the packet p.
$Q^{[\text{tag}_p]}$	The queue associated to the packets with the tag tag_p.
$\mu_{(i,j)}$	Capacity of the link (i, j) or the amount of data that can be transferred over it in the current time slot.
c	The network capacity vector or the set of all the links capacities $\left(\mu_{(i,j)\epsilon L}\right)$ in the network.
Γ	The network capacity region defined by the collection of all possible network capacity vectors c.

The routing problem, investigated in this paper, consists then in performing routing control actions in a way to (1) maximize data transfer during the current time slot, (2) ensure an end-to-end delay performance in per-traffic flow basis, depending on the urgency of each traffic flow.

B. BP—The Legacy Version

Unlike classical routing protocols, BP routing does not perform any explicit path search from source to destination. Instead, as illustrated in Algorithm 1, the routing decisions are made independently for each packet by solving, at each time slot, two principal problems at two different levels. First,the $flow\ selection$ problem at the local level (i.e., at the node level), second $link\ scheduling$ problem occurring at the global level (i.e., at the network level).

1) Flow Selection or Packet Selection: At this phase, the aim consists in deciding for each link $(i, j)\epsilon L$ (cf. Table I), which traffic flow (i.e., packet) is candidate for the next forwarding

Algorithm 1: Backpressure, the legacy version.

```
▷ $Step\,1$ : Packet selection
for all $links\ (i,j) \in \mathcal{L}$ do
    for all $flows\ f_c \in \mathcal{F}_{(i,j)}$ do
        $\mathcal{W}^{f_c}_{(i,j)} \leftarrow Q^{f_c}_i - Q^{f_c}_j$
    end for
    $\mathcal{W}^*_{(i,j)} \leftarrow max_{f \in \mathcal{F}_{(i,j)}} W^f_{(i,j)}$
    $f^*_{(i,j)} \leftarrow arg\,max_{f \in \mathcal{F}_{(i,j)}} W^f_{(i,j)}$
end for

▷ $Step\,2$ : Link scheduling
for all $c \in \Gamma$ do
    $Sum_c \leftarrow \sum_{l \epsilon L} \mu_l \mathcal{W}^*_l$
end for
$c^* \leftarrow arg\,max_{c \in \Gamma} Sum_c$

▷ Data transfer based on the selected $c^*$
for all $(i,j) \in \mathcal{L} : (Q^{f^*_{(i,j)}}_i - Q^{f^*_{(i,j)}}_j) > 0$ do
    $Transfer\ \mu^{c^*}_{(i,j)}\ data\ units\ of\ f^*_{(i,j)}\ from\ Q_i\ to\ Q_j$
end for
```

operation. To do so, first, each node $i \,\epsilon\, N$ computes for each outgoing link a weight as a function of a local per-flow[1] queue $Q^{f_c}_i$. For a given flow f_c and a link (i,j), we denote by $W^{f_c}_{(i,j)}$ the weight, also referred to as *BP*, such that

$$W^{f_c}_{(i,j)} = Q^{f_c}_i - Q^{f_c}_j. \tag{1}$$

Next, based on the computed weights, the flow f^* that maximizes the local queue differential, i.e., the flow having the maximum link weight $W^*_{(i,j)}$, is selected. The computed $W^*_{(i,j)}$ is, therefore, expressed as follows:

$$W^*_{(i,j)} = \max_{f \epsilon F_{(i,j)}} W^f_{(i,j)}. \tag{2}$$

2) Link Scheduling: In the second phase of the legacy BP, a set of links are selected to be activated simultaneously among the list of all nonconflicting links in the network [4]. Scheduling a set of links to transmit concurrently is, therefore, a matter of selecting one link capacity vector $c^* \epsilon \Gamma$ that satisfies the (3), where the weight W_l refers to the maximum link weight W^*_l computed at the previous phase.

$$c^* = \arg\max_{c \epsilon \Gamma} \sum_{l \epsilon L} \mu_l W_l. \tag{3}$$

As a final step, and for each link $l \,\epsilon\, L$, a transmission rate μ_l is offered to the corresponding flow f^*. Notice that, f^* refers to the flow selected over the link l during the previous phase (flow selection). We refer the reader to [3] for an accurate mathematical analysis of the BP algorithm and to [6] for further details about it in addition to some illustrative examples.

C. Overcoming the Delay Limitation of the BP

The previous legacy version of BP has been proven mathematically to stabilize the network, i.e., maintains finite queues at every time instant. Packets were able to find their way by simply moving in the direction of the decreasing backlog [7]. However, such simple strategy can often lead to a significant large latency due to the following reasons [15].

1) First, the BP suffers from the so-called *slow start phenomena*, that is, in case of a slightly loaded network, packets may take unnecessarily long routes. While this extensive exploration is essential when the network is heavily loaded in order to maintain its stability (i.e., load balancing over the whole network), under light or moderate loads the situation can lead to a significant QoS deterioration, especially in terms of end-to-end delay.
2) Second, *the fluctuation* in terms of queue backlog eventually leads to routing-loops formation causing long packet delays [16].
3) Third, the so-called *last packet problem*. This issue is basically caused by the absence of consistent BP toward the destination, which is the case with low-rate flows or short-lived injected packets [10], [12].

The above-mentioned delay inefficiency of BP was the key driver behind several works in the literature.

To overcome the slow start phenomena, Neely *et al.* added a new constant shortest path bias (parameterized by a per-link cost B) to the BP calculation [7]. Nodes are, therefore, willing to route packets in the direction of their destinations [14]. Ying *et al.* [5], however, improved the achieved delay by proposing a joint traffic-splitting (parametrized by a parameter K) and shortest-path-aided BP routing protocol. In this scheme, the per-destination queues of the legacy BP are replaced by a new hop-queues structure, where each node maintains the same queue Q_h for the packets to be delivered to a destination within h hops. The objective is to minimize the average number of hops per packet delivery, thereby minimizing the end-to-end propagation delay. As shown in Table II, one potential challenge for these works is that their performance depends to a large extent on the choice of the parameters K and B. In fact, this creates another challenge since the optimal values of these parameters vary depending on the traffic load, which is difficult to predict in advance.

To reduce packet loops, Alresaini *et al.* [10] propose to introduce redundant packets to build up gradient toward destinations in a faster way. Following the same intuition, a new shadow queuing architecture is proposed in [11] wherein, instead of redundant packets, a fictitious queuing system in addition to a new per-neighbor physical queues at each node (instead of per-destination queues) are designed. A common challenge for all the above schemes is that they are built upon the assumption of a fixed routing scenario, i.e., the route for each flow is chosen upon arrival using some standard wireless multihop networks routing algorithms and the BP algorithm is simply used to schedule packets transmission [7].

Other works focused on alleviating the last packet problem, such as [12] in which the authors proposed Delay-based Back-Pressure (D-BP) that uses the sojourn-time difference as a new metric instead of the queue-length-based BP. As the previous works, the main drawback of this scheme is the fact that it is destined to wireless multihop networks with precalculated route too. Huang and Neely [13] used a virtual backlog process

[1] Note that the classical BP assumes that each node i maintains a separate queue $Q^{f_c}_i$ for each per destination flow f_c.

TABLE II
SUMMARY AND COMPARISON OF THE DESCRIBED BP-BASED SCHEMES WITH ENHANCED DELAY PERFORMANCE

Scheme	Architecture	Queues management strategy	Key principle	Limitations
Joint routing and power allocation BP [7]	Distributed	Per-flow	Incorporating an additional progress-to-destination metric in BP calculation.	Performance depends on the choice of internal parameters, which are difficult to predict in advance.
Shortest-path aided BP [5]	Centralized	Per-hop		
DVV-BP [8], [9]	Distributed	Per-node	No consideration for the urgency of each flow.	
BP with adaptive redundancy [10]	Distributed	Per-flow	Introducing redundant packets when queue occupancy is low	Greater number of transmissions due to the introduced redundancy.
BP with shadow queueing [11]	Centralized	Per-neighbor	Using shadow packets (as counters) instead of redundant packets.	Delay enhancement at the expense of the throughput degradation, in addition to the precomputed routes assumption.
Delay-based BP [12]	Centralized	Per-flow	Using the sojourn-time difference instead of the queue-length-based BP.	Precomputed routes assumption.
Fast quadratic lyapunov based BP [13]	Centralized	Per-flow	Introducing a virtual backlog mechanism with LIFO service discipline	An arbitrary fraction of packets to be dropped.
General queue-dependent BP [14]	Distributed	Per-flow	Incorporating “beyond one hop” queue state information in BP calculation.	The need for additional overhead to get information beyond one hop, which increases the network load.

with last in last out (LIFO) service discipline on top of the legacy BP. In fact, this introduced idea improved the order of the utility-delay tradeoff, but at the expense of an arbitrary fraction of packets to be dropped [14]. More recently, a new class of enhanced BP-based algorithms was introduced in [14]. This class incorporates the global queue state information (beyond one hop) instead of the only one-hop queue size difference used in computing the BP values. However, this solution implies an additional overhead induced by the mechanism used to get such information beyond one hop, which leads to an increasing network load.

Finally, the works proposed in [8] and [9] combine both delay and queue complexity reduction for regular (i.e., grid-like) wireless mesh topology (referred to as Distributed Variable-V Backpressure, or DVV-BP, in Section IV). Both schemes proposed an interesting control mechanism based on the Lyapunov drift-plus-penalty optimization framework introduced in [17]. This mechanism consists in adding an additional progress-to-destination term, based on Geo-location, to the (1), which enables a better tradeoff between two types of routing decisions. Routing decisions that aim to achieve network stability (i.e., keep the queues under control to prevent any overflow) and those aiming to reach near-optimal values for some objective performance metrics (e.g., end-to-end delay). The queue complexity is reduced in both schemes by maintaining a single queue in each node, instead of a per flow/destination queue as in the original BP. However, despite its attractive low queues complexity, and similar to previous works, there is no consideration for the urgency of each traffic flow separately. The key principles and limitations of each of the above-mentioned works are summarized in Table II.

While not being the firsts attempting to enhance the bP delay performance, as discussed previously, the motivation and design aspects behind our proposal differ significantly from the aforementioned works. Apart from the aim for a distributed scheme and a queue management with low complexity, the main driver of our proposal is to enable differentiation between flows based on the urgency level of each flow. The proposed strategy prioritizes flows with strict end-to-end delay requirements, without sacrificing flows with less strict end-to-end delay requirements. A detailed description of our proposal is provided in Section III.

III. LIFETIME-AWARE BP

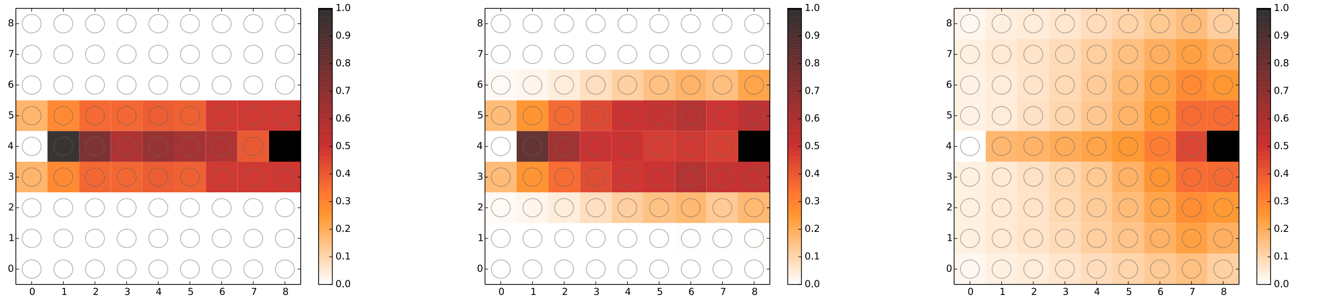
Our proposal is based on the drift-plus-penalty framework introduced in [17]. In this framework, instead of calculating the weight as in (1), a new component is added as follows:

$$W_{(i,j)} = (Q_i - Q_j) - V * P(i,j,d) \tag{4}$$

where $(Q_i - Q_j)$ is the difference in queue backlog between nodes i and j. The additional term $P\,(i,j,d)$ is used to minimize the distance covered by a packet from the source node to the destination d, and is computed as the cost of traversing the link (i,j) in the path leading to d. The framework involves taking routing actions to minimize the cost function, subject to maintaining the stability of queues in the network. For instance, the following function is the penalty function used in [8]:

$$P(i,j,d) = \begin{cases} +1, & j \text{ farther to } d \text{ than } i \\ -1, & j \text{ closer to } d \text{ than } i. \end{cases} \tag{5}$$

This penalty function is weighted by $V \geqslant 0$, where V is a parameter representing how much we emphasize the penalty minimization. As illustrated in Fig. 2, V enables a tradeoff between the queue-based component $(\triangle Q_{ij})$ and the penalty function $P(i,j,d)$. A low valuefor V means a high emphasis on queues stability $(\triangle Q_{ij})$ and so a more load balancing oriented BP. In



(a)

(b)

(c)

Fig. 2. Heatmaps illustrating the degree of traffic distribution as a function of V, in a 9×9 grid network where nodes are labeled from (0, 0) to (8, 8). (a) High V. (b) Medium V. (c) Low V.

the opposite case, a high value to V means a high emphasize on minimizing the distance toward the destination d (penalty function component), which is reflected by the selection of the shortest path in Fig. 2.

The intuition behind our contribution is based on the previous control mechanism. The main idea consists in adding a prerouting phase prior to performing any computation and choosing the next hop. In this phase, we *categorize the traffic flows in terms of end-to-end delay requirements* and *compute the V value, in per-packet fashion, depending on the age of the packet and the urgency level of its corresponding flow*. To the best of our knowledge, this is the first work, which leverages the parameter V in prioritizing traffic flows based on their end-to-end delay requirements.

The overall proposal is organized into two phases, namely: *Prerouting calculation phase*, and *the Routing decision phase.* In the following, we will provide a detailed overview on each phase.

A. Prerouting Phase

In this phase, one urgent packet is selected among the enqueued ones and assigned a value of V according to its urgency. To achieve this, four basic steps need to be undertaken, as illustrated in Algorithm 2 and explained in the following.

1) Packet Queuing: we assume that the injected traffic flows are categorized based on their end-to-end delay requirements. Each packet p is, therefore, tagged with tag_p depending on the category of its flow. For scalability concerns, we limit the number of categories to four, each with a specified tags, namely: *"High+"* for very high (critical) urgency, *High* for high urgency, *Med* for flows with medium urgency, and *Low* for low urgency (see Fig. 3). Any arriving packet p is placed in one of the available four queues, corresponding to the above defined categories. A LifeTime value is associated with each queue and used as a deadline, after which the information conveyed by the queued packets starts to lose its utility. In other words, whenever possible, the age of a given packet p (Age_p) should not exceed the LifeTime of its queue $Q^{[\text{tag}_p]}$. Where Age_p refers to the cumulative delay from the source node to the current node.

2) Urgency Calculation: Let HoLs be the set of packets at the Head of Line position in the four previous queues. In this step, an Urgency value is calculated for each packet $P \in \text{HoLs}$.

```
Algorithm 2: Pre-Retouing Phase.
Input: $Packet\ p$
Output: $Parameter\ V$

▷ $Step\ 1$ : En-queuing $p$
$Tags \leftarrow \{High+, High, Med, Low\}$
$tag_p \leftarrow ExtractTag(p)$
$Q^{[tag_p]}.enqueue(p)$

▷ $Step\ 2$ : Urgency calculation for each packet at the head of line position in $Q^{[tag]}$, $tag \in Tags$
$HoLs \leftarrow \{HoL :\ HoL = Q^{[tag]}.getHoL(),\ tag \in Tags\}$
for all $packets\ HoL \in HoLs$ do
    $tag_{HoL} \leftarrow ExtractTag(HoL)$
    $Urgency_{HoL} \leftarrow Age_{HoL}/LifeTime_{Q^{[tag_{HoL}]}}$
end for

▷ $Step\ 3$ : $P^*$ Selection
$P^* \leftarrow arg\,max_{HoL \in HoLs} Urgency_{HoL}$

▷ $Step\ 4$: $V$ Calculation
$V_{P^*} \leftarrow Q_{max} * Urgency_{P^*}$

return $V_{P^*}$
```

As illustrated in (6), the Urgency of a packet P depends on the ratio between its Age: Age_P, and the LifeTime of its associated queue $Q^{[\text{tag}_P]}$

$$\text{Urgency}_P = \frac{\text{Age}_P}{\text{LifeTime}_{Q^{[\text{tag}_P]}}}. \tag{6}$$

3) P Selection:* As shown in Fig. 4, one packet $P^* \in \text{HoLs}$ is selected in this step, based on the Urgency values calculated previously. Using the formula in (7), the most urgent packet in HoLs is selected as follows:

$$P^* = \arg\max_{P \in \text{HoLs}} \text{Urgency}_P. \tag{7}$$

4) V Calculation: This step consists in calculating a new value for the routing parameter V, using the maximum queue length $Q_{\max}$, in a way to reflect the urgency of the selected packet P^*:

$$V(P^*) = Q_{\max} \cdot \text{Urgency}_{P*}. \tag{8}$$

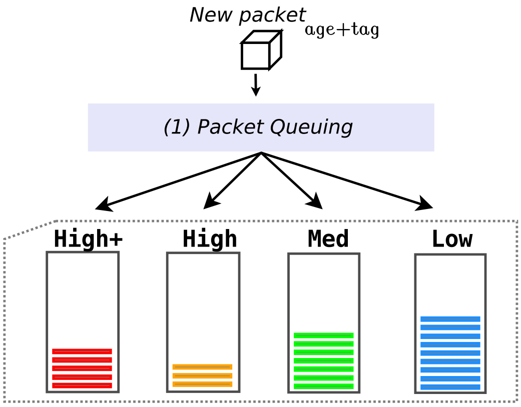


Fig. 3. Illustration of packets queuing process in the defined four categories.

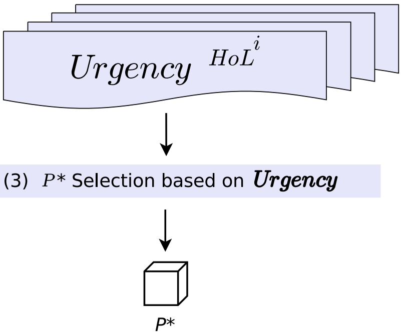


Fig. 4. Packet P^* selection based on the calculated Urgency values.

To better understand the intuition behind (8), let us discuss more about the variation of V as a function of the packet's age. To this end, we identify three cases as follows.

1) $\text{Age}_{P*} \ll \text{Lifetime}_{Q[\text{tag}_{P*}]}$: In this case, the age of the packet is too small compared to the corresponding LifeTime. Based on (6) Urgency_{P*} is assigned a low value, which implies a low value for $V(P^*)$ based on (8). The penalty component consequently, is given less importance when calculating the next hop according to (4). This is due to the fact that there is no imminent necessity to transmit the packet P^*, and instead the priority is to ensure queues stability through load balancing.
2) $\text{Age}_{P*} < \text{LifeTime}_{Q[\text{tag}_{P*}]}$: As soon as the age of a packet approaches its LifeTime, we gradually assign a higher value to V. In contrast to the previous case, by assigning higher value to V, we give more weight to the penalty function component, and therefore, we direct the packets toward a closer to destination neighbor, among all one hop neighbors.
3) $\text{Age}_{P*} \geq \text{LifeTime}_{Q[\text{tag}_{P*}]}$: At this stage, the ultimate priority is to reach the destination as soon as possible. To do so, the next node should compulsory be a closer neighbor to the destination. To enforce such a choice, the routing calculation for the current packet must be based only on the penalty function component. Since the load balancing component $\triangle Q_{ij}$ will never exceed $Q_{\max}$ $(\forall(i,j)\epsilon L : \triangle Q_{ij} \leqslant Q_{\max})$, the calculated V must, therefore, be greater or equal to $Q_{\max}$.

TABLE III
SIMULATION PARAMETERS

Simulation Time	50 s
Traffic generators	CBR
Transport layer	User datagram protocol (UDP)
Packet size	1500 bytes
Data queue size	400 packets
HELLO period	100 ms
MAC layer	IEEE 802.11a
Wireless band	5 GHz
Data rate	54 Mb/s
Propagation model	Constant speed propagation delay model

B. Routing Decision Phase

The objective of the previous phase is achieved by the selection of the most urgent HoL packet. The selected packet is assigned a high, medium or low value of V depending on its urgency. Since the value of V has a direct impact on the forwarding process (see Fig. 2), the key idea of this phase is to exploit these results in the selection of the next hop.

As discussed previously, selecting the next hop is made based on (4). To forward the selected packet, a node i calculates the value $W_{(i,j)}$ for each neighbor node j, then the node with the maximum BP value W^* is chosen as next hop in the routing path. Depending on the urgency of the packet, the selected node could be a closer to the destination, among node is neighbors (in case of an urgent packet, i.e., with a high V), or the less congested node (in case of a nonurgent packet, i.e., a lower V). The overall view of our system including the different steps involved in the proposed scheme is shown in Fig. 5.

IV. PERFORMANCE EVALUATION

In this section, we validate our proposal by conducting a set of simulation experiments, collecting the results and analyzing them in detail. First, we briefly present the simulation setup and the chosen evaluation parameters. Then, we illustrate how the different internal parameters can affect the total behavior of our system. Finally, we present a comparative experiment in which we compare the performance of our proposal against another state-of-the-art BP-based routing protocol (i.e., the work presented in [8] and [9] and discuss the results.

A. Simulation Setup

We have conducted all the simulations using ns-3 [18]. As listed in Table III, the duration of each simulation is set to 50 s. The simulated network is a grid backhaul of wireless nodes (See Figs. 6 and 10), where each node is equipped with a single IEEE 802.11a WiFi interface configured to operate on the same channel at a link rate of 54 Mpbs, as well as the same carrier sense ranges.

To exchange information about queues size, we used a decentralized method that solely requires HELLO based communication between neighboring nodes. For optimal operation, each node is required to exchange HELLO messages at a constant rate of 10 packets per second (i.e., a period of 100 ms).

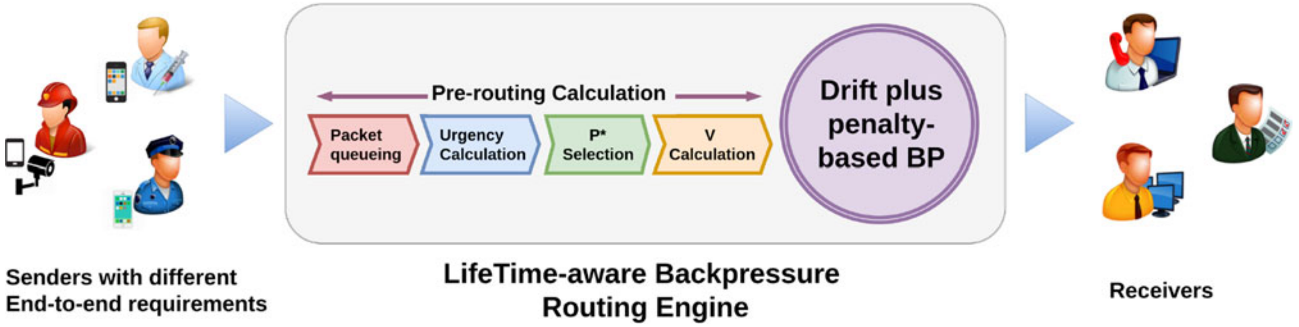

Pre-routing Calculation
Packet queueing
Urgency Calculation
P* Selection
V Calculation
Drift plus penalty-based BP
Senders with different End-to-end requirements
LifeTime-aware Backpressure Routing Engine
Receivers

Fig. 5. Overview of the system.

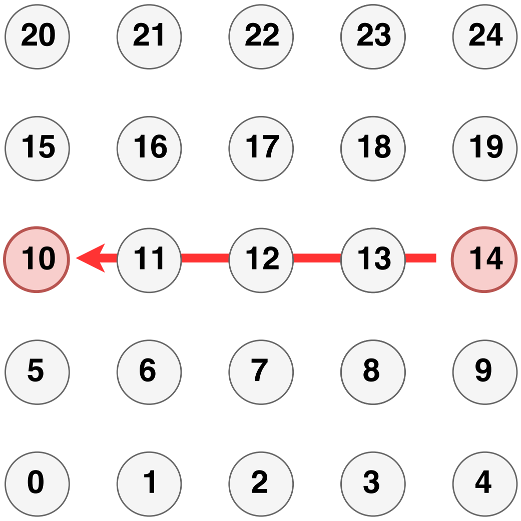


Fig. 6. Topology used in the illustrative experiment, where a single UDP flow is sent from node 14 to node 10.

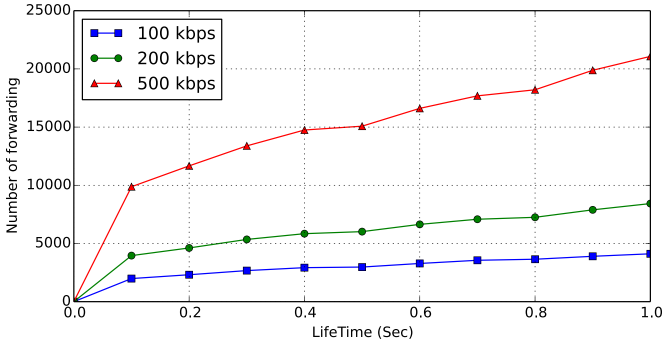


Fig. 7. Impact of the LifeTime value on the total number of packets forwarding.

TABLE IV
EVALUATION METRICS

Metric	Definition
End-to-end delay	Average time taken to deliver a packet (urgent or nonurgent) from the sender to the receiver.
Throughput	The amount of data successfully delivered in a unit of time.
PDR	Ratio of received packets over the total number of packets sent.

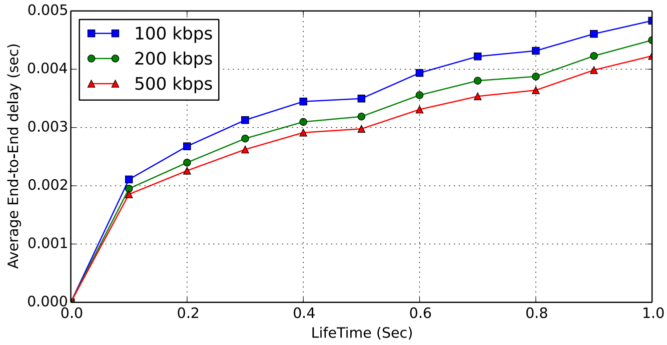


Fig. 8. Impact of the LifeTime value on the achieved average end-to-end delay.

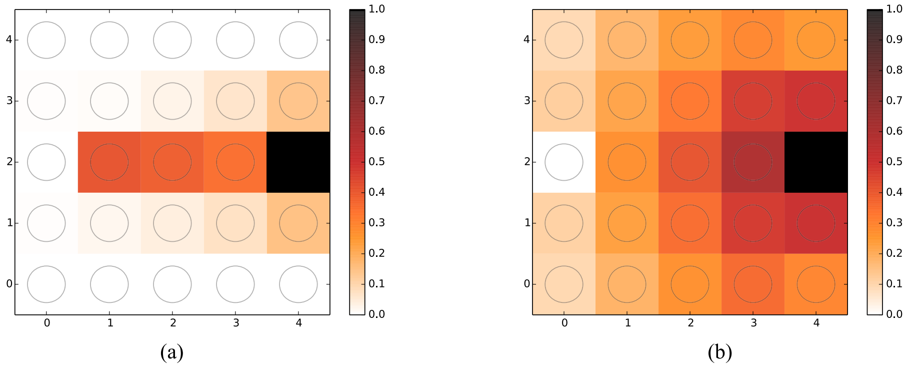
Table IV summarizes the network performance metrics used in our evaluation.

B. Illustrative Experiments

Experiment 1: In order to illustrate the effect of using different LifeTime values on the overall behavior of our system, we focus in this experiment on one generic queue corresponding to one traffic flow category. The 5×5 grid topology used in this experiment is shown in Fig. 6. A constant bit rate (CBR) traffic flow is sent from node 14 to the node 10 using different input rates (i.e., 100 kb/s, 200 kb/s, and 500 kb/s). We repeat the same experiment 10 times with an increase in the LifeTime by 100 ms each time, up to 1000 ms (1 sec).

The plotted results in Fig. 7 reveal the impact of incrementing the LifeTime value on the total number of forwarding decisions. Under the same input rate, we notice that as long as we increment the LifeTime value, forwarding decisions are made more frequently. This effect is more apparent in case of higher input rate (i.e., 500 kb/s) since more packets are injected, and hence, more forwarding decisions are made.

The reason behind this correlation is the close relationship in our control mechanism between 1) The LifeTime of a queue and the urgency calculation of its enqueued packets. 2) The urgency of the packet to be forwarded and the load balancing tendency of the system. In fact, according to (6), by incrementing the LifeTime value of a queue, we are proportionally penalizing the enqueued packets by decreasing their urgency. This means a low value for V according to (8), and therefore, a tendency of the system to enlarge the amount of explored routes (i.e.,



(a)

(b)

Fig. 9. Packets distribution heatmaps for (a) Hight+ urgency traffic flow with LifeTime = 1 ms. (b) Low urgency traffic flow with LifeTime = 10 s.

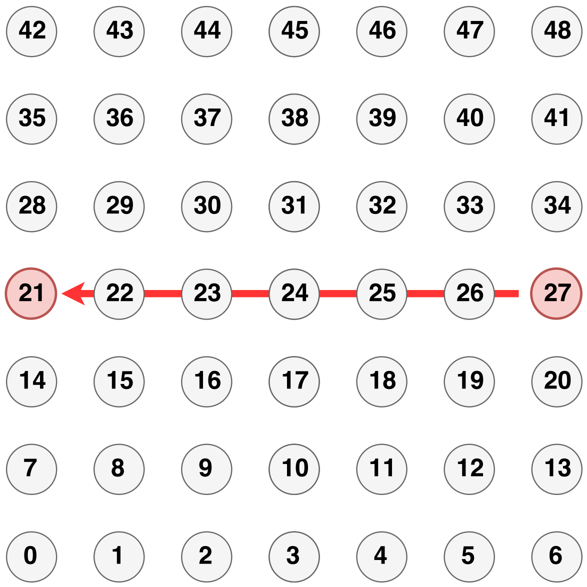


Fig. 10. 7 × 7 grid topology used in the third experiment: from the same node, two traffic flows are sent, each time using a different pair of LifeTime values ($\text{LifeTime}_{\text{High}+}$, $\text{LifeTime}_{\text{Low}}$).

more load balancing). Such behavior means two things; first the number of forwarding actions will grow proportionally (since more nodes are visited), second, the corresponding end-to-end delay will augment by the same pace. This latter conclusion about the delay is confirmed by the results shown in Fig. 8.

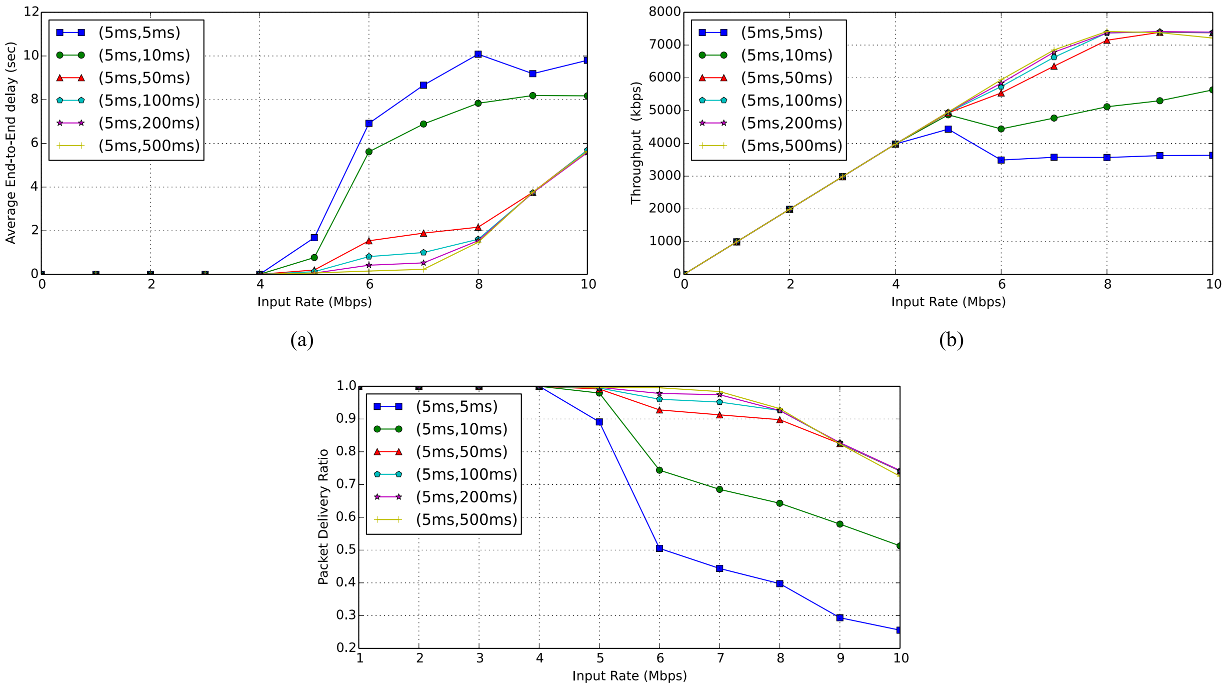
Experiment 2: This experiment aims to confirm our previous analysis by examining the behavior of our system when dealing with heterogeneous traffic flows. The experiment is conducted using the same network topology shown in Fig. 6. Using the same pair of nodes (i.e., node 14 as source and node 10 as a destination) we inject two traffic flows with different requirements in terms of urgency.

We use heatmaps to examine the packets distribution in both cases as depicted in Fig. 9. A heatmap is visually easy to interpret since it is color coded. Here, the degree of darkness visually reflects the number of forwarding decisions made at a specific node. A node forwarding a large number of packets is represented by black squares [e.g., the source node 14 at the position (4, 2)] and a node with a smaller value is represented by a lighter squares [e.g., node 20 at the position (0,4)]. The heatmap showing all the nodes gives us an idea about how the forwarding is performed from a global perspective.

Fig. 9 confirms our prior analysis as we clearly notice the different forwarding patterns for each traffic flow. Fig. 9(a) depicts the packets distribution of the Hight+ urgency traffic flow, in which we observe that a few nodes only are involved in the forwarding process. In contrast, Fig. 9(b) shows that almost all nodes are involved in forwarding Low urgency traffic flow packets. Such behavior is the reason behind the results obtained in the previous experiment. Using different values for LifeTime intends to enable load balancing at different degrees. In case of Hight+ urgency traffic flow, the load balancing is not allowed, except of the source node's neighbors (i.e., the nodes 9, 8, 13, 18, 19 shown in Fig. 6) where we can see some sort of load balancing. Fig. 9(b), however, illustrates the opposite since the constraints are softer here, thus, the load balancing is more likely to be allowed at a higher degree.

Experiment 3: In a grid topology of 7 × 7 nodes (see Fig. 10), we repeat the previous experiment using different pairs of LifeTime values. The aim is to analyze the effect of using close (adjacent) LifeTime values for both traffic flows [e.g., (5 ms, 5 ms) meaning that 5 ms as a LifeTime for High+ urgent flow and 5 ms for the other one] or more distant values such as (5 ms, 200 ms) or (5 ms, 500 ms). This effect is analyzed by gradually increasing the input rate up to 10 Mb/s. Fig. 11 shows the results for urgent traffic flows in terms of the achieved average end-to-end delay, throughput and packet delivery ratio (PDR).

As shown in Fig. 11, the use of distant LifeTime values for urgent/nonurgent traffic flows allows urgent traffic to experience a better QoS up to a certain limit. The reason behind such performance is the load balancing-based control mechanism previously discussed in experiment 2. This mechanism relaxes the congested nodes (i.e., nodes, which are busy in forwarding urgent traffic) by pushing away less urgent packets, thereby urgent packets are delivered with lower end-to-end delay and higher throughput. From these results we conclude that the difference (i.e., the gap) between the chosen LifeTime values plays a key role in determining the achieved performance. The bigger the difference between the two values, the higher QoS the urgent traffic will experience. However, at a certain level, this difference becomes less apparent since the load balancing is limited physically by the size of the network.

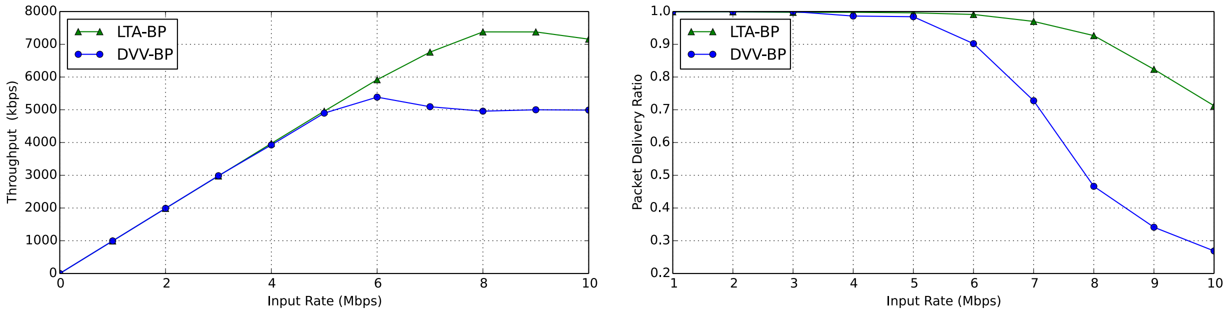


(a)

(b)

(c)

Fig. 11. Network performance for urgent traffic flows under different input rates and using different pairs of LifeTime values. (a) Average end-to-end delay. (b) Throughput variation. (c) PDR.



(a)

(b)

Fig. 12. Network performance for urgent traffic flows under different input rates: Our scheme (LTA-BP) versus dynamic variable V scheme (DVV-BP). (a) Throughput variation. (b) PDR variation.

C. Comparative Experiment

In this experiment, we compare the performance of our scheme with variable V scheme proposed in [9] (referred to here as DVV-BP). In the same grid topology (see Fig. 10), we send the same traffic flows with the LifeTime pair (5 ms, 500 ms), and repeat the simulation several times with an increase of the input rate by 1 Mb/s each time, up to 10 Mb/s. Results are obtained by averaging values from 100 runs with different seeds using the high performance computing platform IBNBADIS.[2]

Figs. 12–14 depict the obtained results in terms of end-to-end delay, throughput and PDR. The results plotted in Figs. 12 and 13, highlight the superiority of our scheme over DVV-BP, under different evaluation metrics. In terms of throughput and starting from an input rate of 5 Mb/s, our scheme clearly outperforms DVV-BP with an incrementing throughput up to 7.3 Mb/s. The same trend is observed for the PDR where we observe a sharp degradation for DVV-BP performance, especially under higher input rate values. In contrast, to avoid such QoS deterioration, our scheme as discussed previously, tends to alleviate the congestion at the shortest paths by pushing away less urgent packets to less congested nodes.

In terms of end-to-end delay, both schemes perform well under input rates lower than 5 Mb/s although our scheme is slightly better than DVV-BP when the input rate is between 3 and 5 Mb/s, as shown in Fig. 13. Beyond 5 Mb/s, the average end-to-end delay tends to increase under both schemes but withdifferent paces. In DVV-BP, the achieved delay exhibits sharp increase till it reaches a peak value of 5.5 s after which it stabilizes, whereas in our scheme the delay shows a less aggressive increase till it attains the same peak value when the

[2] High Performance Computing Platform IBNBADIS (www.ibnbadis.cerist.dz) provided by the Research Center on Scientific and Technical Information—CERIST (Algeria).

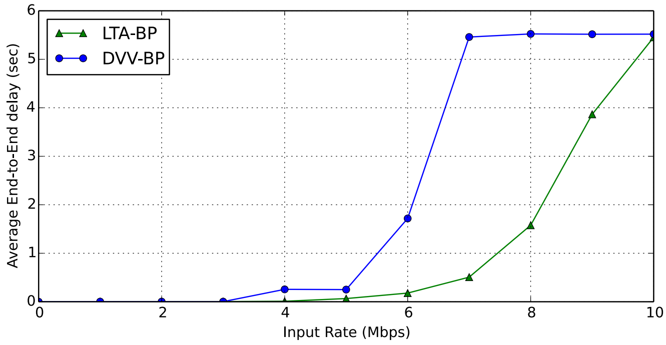


Fig. 13. Average end-to-end delay for urgent traffic flows (High+ urgency) under different input rates: Our scheme (LTA-BP) versus dynamic variable V scheme (DVV-BP).

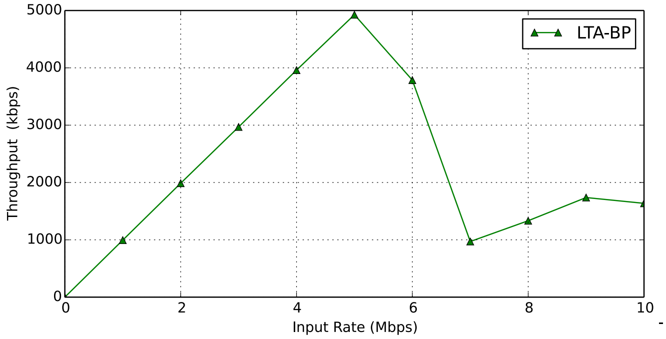


Fig. 14. Throughput variation for less urgent traffic flows (low urgency) under different network loads.

input rate reaches 10 Mb/s. This performance degradation is due to the same reason discussed above; as reducing the load at the shortest path allows urgent flows to reach their destination faster, especially in highly loaded scenarios. However, under 10 Mb/s input rate, the network reaches its full capacity (i.e., the sum of 10 Mb/s for each traffic flow), and thus, traffic flows suffer from high end-to-end delay in both schemes.

Fig. 14 depicts the throughput variation of less urgent traffic flows when using our scheme. By inspecting the results shown in this figure, we reveal that up to 5 Mb/s of input rate the achieved throughput is proportional to the input rate value. However, beyond 5 Mb/s of input rate, this type of traffic flow suffers from a severe deterioration, almost 80% decrease, of the throughput (from 5 to 1 Mb/s). Despite this severe decrease, resulting from delaying these flows in order to offer a better QoS to traffic flows with more strict delay requirements (high+ urgency), our scheme do not fall in an endless starvation situation of less urgent traffic flows. Notice that a starvation situation or state refers to the undesirable state in which a specific traffic flow receives zero or close-to-zero throughput [19]. As illustrated in Fig. 14, and even in highly loaded situations, our scheme is able to protect these flows from total starvation by ensuring a minimum throughput of 1 to 2 Mb/s. Such minimum throughput is the result of the novel approach of calculating the urgency in our scheme. The approach consisted in considering any packet, regardless of its type, as urgent as soon as it approaches its deadline. Therefore, since every packet belonging to a less-urgent traffic flow will get closer to its deadline at a certain time point, it is automatically treated as urgent packet and the forwarding process acts upon this consideration while selecting the next hop.

V. Conclusion

In this paper, we dealt with the delay inefficiency issue of dynamic BP family protocols and proposed a new scheme, dubbed LifeTime-Aware BackPressure (LTA-BP), to overcome this limitation. The novelty of our scheme lies in its ability to adjust the routing pattern depending on the delay requirement of each traffic flow, thanks to our urgency calculation approach. The ultimate goal of LTA-BP is to offer an enhanced QoS for real-time applications with strict end-to-end delay requirements while avoiding the starvation of other applications with eventually softer end-to-end delay requirements. We conducted a set of experiments using ns-3 simulator running on the high performance computing platform IBNBADIS, and the obtained results have demonstrated the efficiency of LTA-BP and its supremacy over another state-of-the-art scheme. As a future work, we plan to extend our scheme to accommodate extremely strict requirements of traffic flows carrying various data types in real-word disaster response scenarios. In such scenarios, the QoS level offered to each packet depends on several factors related to the criticality of the information carried in the packet, the physical location of the sender, etc.

References

[1] U. A. C. Command, "A request for information/notice investigation for cellular Long Term Evolution (LTE) technologies," U.S. Department of the Army, Arlington, VA, USA, Tech. Rep. W56KGU14RA012, 2014.

[2] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: A survey," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 102–110, Feb. 2016.

[3] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[4] J.-Y. Yoo, C. Sengul, R. Merz, and J. Kim, "Backpressure scheduling in ieee 802.11 wireless mesh networks: Gap between theory and practice," *Comput. Netw.*, vol. 56, no. 12, pp. 2934–2948, 2012.

[5] A. R. Lei Ying, S. Shakkottai, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun. 2011.

[6] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 279–290.

[7] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. vol. 23, no. 1, pp. 89–103, Jan. 2005.

[8] J. Núñez-Martínez and J. Mangues-Bafalluy, "Distributed lyapunov drift-plus-penalty routing for wifi mesh networks with adaptive penalty weight," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, 2012, pp. 1–6.

[9] J. Núñez-Martínez, J. Mangues-Bafalluy, and J. Baranda, "Anycast backpressure routing: Scalable mobile backhaul for dense small cell deployments," *IEEE Commun. Lett.*, vol. 17, no. 12, pp. 2316–2319, Dec. 2013.

[10] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Backpressure with adaptive redundancy (bwar)," in *Proc. IEEE INFOCOM*, 2012, pp. 2300–2308.

[11] L. X. Bui, R. Srikant, and A. Stolyar, "A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm," *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1597–1609, Dec. 2011.

[12] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, Oct. 2013.

[13] L. Huang and M. J. Neely, “Delay reduction via lagrange multipliers in stochastic network optimization,” *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 842–857, Apr. 2011.

[14] Y. Cui, E. Yeh, and R. Liu, “Enhancing the delay performance of dynamic backpressure algorithms,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 954–967, Apr. 2016.

[15] L. A. Maglaras and D. Katsaros, “Delay efficient backpressure routing in wireless ad hoc networks,” *EAI Endorsed Trans. Mobile Commun. Appl.*, vol. 14, no. 1, pp. 1–6, Sep. 2014.

[16] A. Rai, C.-P. Li, G. Paschos, and E. Modiano, “Loop-free backpressure routing using link-reversal algorithms,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2988–3002, Oct. 2017.

[17] L. T. Leonidas Georgiadis and M. J. Neely, *Resource Allocation and Cross-Layer Control in Wireless Networks* (Foundations and Trends in Networking), vol. 1, no. 1, Breda, The Netherlands: Now Publishers, 2006.

[18] T. N. Simulator, “ns-3, version 3.24.4.” 2015. [Online]. Available: https://www.nsnam.org/

[19] C. Kai and S. C. Liew, “Temporal starvation in CSMA wireless networks,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1515–1529, Jul. 2015.

Abdelbaset Kabou received the Magister degree in software and network engineering and Engineering degree in computer science with majors from USTO-MB University, Bir El Djir, Algeria, and the University of Béchar, Béchar, Algeria, in 2007 and 2010, respectively. He is currently working toward the Ph.D. degree at the National Higher School of Computer Science (ESI), Algiers, Algeria.

He is a full-time Researcher with the Research Center on Scientific and Technical Information, Algiers, Algeria. His research interests include wireless networking with a focus on routing, stochastic optimization, and quality of service.

Nadia Nouali-Taboudjemat received the Engineering degree in computer science from the University of Science and Technology Houari Boumediene (USTHB), the Magister degree in computer science from the Advanced Technologies Research Centre (CDTA), Algiers, and the Ph.D. degree in computer science from USTHB.

She is a permanent full-time Senior Researcher with the Research Center on Scientific and Technical Information (CERIST), Algiers, Algeria, where she is leading the Theory and Engineering of Computer Systems Division and the Ubiquitous Systems Group. Her research interests include ubiquitous computing, Big Data, Big Graphs, ICT-based disaster management, smart cities, and sustainable environment.

Soufiene Djahel (GS’08–M’11–SM’16) received the Ph.D. degree in computer science from the Lille 1 University-Science and Technology, Villeneuve-d’Ascq, France, in 2010.

He has been a Senior Lecturer with Manchester Metropolitan University, Manchester, U.K., since September 2015, and was an Engineering Research Manager with the University College Dublin (UCD), Dublin, Ireland, from February 2012 to September 2015. Before joining UCD, he was a Postdoctoral Fellow with the École nationale supérieure d’informatique pour l’industrie et l’entreprise, Essonne, France. His research interests include intelligent transportation systems, security and QoS issues in wireless networks, and E-health.

He serves as TPC member for many IEEE flagship conferences and as a Reviewer for several IEEE journals in his research areas.

Saïd Yahiaoui received the Magister degree in networking and distributed systems in computer science from Abderrahmane Mira University, Bejaia, Algeria, in 2005 and the Ph.D. degree in computer science from Claude Bernard Lyon 1 University, Villeurbanne, France, in 2013.

He is a full-time Researcher with the Research Center on Scientific and Technical Information (CERIST), Algiers, Algeria. From 2007 to 2010, he was a Research Assistant with CERIST. His research interests include wireless networking, parallel and distributed computing, and graph algorithms.

Omar Nouali received the Ph.D. degree in computer science from the University of Science and Technology Houari Boumediene, Algiers, Algeria, in 2004.

He is currently the Director of Research and the Head of the Security Department, Research Centre on Scientific and Technical Information, Algiers. His research interests include information filtering and computer security.